

TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta textilní



DIPLOMOVÁ PRÁCE

Simulační zapojení střídavého pohonu s měničem SIMOVERT

2004

Petr Buben

TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta textilní

Obor: 31 - 06 - T

Zaměření: Řízení technologických procesů

Simulační zapojení střídavého pohonu s měničem SIMOVERT

Simulation connection a/c drive with frequency converter
SIMOVERT

Petr Buben

Vedoucí práce: Doc.Ing. Eva Konečná, Csc.

Konzultant: Ing. Patrik Endler

Rozsah práce:

Počet stran: 49

Počet obrázků: 37

Počet tabulek: 1

Počet příloh: 2 + 1 CD

Anotace:

Diplomová práce se zabývá vytvořením programu na vizualizaci simulačních schémat U/f řízení střídavého pohonu a výsledků měření na reálném pohonu.

Dalším bodem je návrh začlenění programu VizData do výuky elektrických pohonů. Popis sestavení simulačních schémat v prostředí MATLAB/SIMULINK i s návodem pro obsluhu. Je zde také přiblížen princip funkce měniče SIMOVERT a práce s programem SIMOVIS.

Annotation:

This Diploma Thesis deals with creation of program for visualization simulation schemes of U/f control alternating current drive and results of measuring and control real gear.

Next part is project of integration the program VizData into teaching electrical gears. Description of frame simulation schemes in program MATLAB/SIMULINK with instruction for service. Here is approximate using principle of frequency converter SIMOVERT and working with program SIMOVIS.

Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 o právu autorském, zejména § 60 (školní dílo) a § 35 (o nevýdělečném užití díla k vnitřní potřebě školy).

Beru na vědomí, že TUL má právo na uzavření licenční smlouvy o užití mé práce a prohlašuji, že **s o u h l a s í m** s případným užitím mé práce (prodej, zapůjčení apod.).

Jsem si vědom toho, že užít své diplomové práce či poskytnout licenci k jejímu využití mohu jen se souhlasem TUL, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše).

Prohlašuji, že jsem diplomovou práci vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce a konzultantem.

V Liberci dne 14. 5. 2004

Podpis:

OBSAH

1	ÚVOD.....	1
2	VYSVĚTLENÍ ZÁKLADNÍCH POJMŮ.....	2
2.1	System.....	2
2.2	Model.....	2
2.3	Modelování.....	3
2.4	Simulace.....	3
3	FUNKCE FREKVENČNÍHO MĚNIČE SIMOVERT.....	4
3.1	Měniče pro regulované pohony.....	4
3.1.1	Nepřímé měniče frekvence s napěťovým meziobvodem.....	4
3.1.2	Přímé měniče frekvence – cyklokonvertory.....	4
3.2	Základní rysy současných měničů SIMOVERT firmy SIEMENS.....	5
4	ŘÍZENÍ ASYNCHRONNÍCH MOTORŮ.....	8
4.1	Skalární řízení.....	9
4.1.1	Frekvenčně proudové řízení.....	9
4.1.2	Frekvenčně napěťové řízení.....	10
4.2	Vektorové řízení.....	11
4.2.1	Přímá metoda vektorového řízení.....	12
4.2.2	Nepřímá metoda vektorového řízení.....	13
4.2.2.1	Nepřímá metoda s čidlem rychlosti rotoru.....	13
4.2.2.2	Nepřímá metoda s čidly napětí.....	13
5	PROGRAM MATLAB/SIMULINK A VYTVOŘENÉ BLOKY.....	14
5.1	Modelování skalárního řízení měniče SIMOVERT.....	17
5.2	Blok U/f CHARAKTERISTIKA.....	19
6	NÁVRH LABORATORNÍHO CVIČENÍ.....	21
6.1	Simulační část.....	21
6.1.1	Model U/f řízení pro textilní aplikace měniče SIMOVERT.....	21
6.1.2	Model U/f řízení měniče SIMOVERT.....	28
6.1.3	Model U/f řízení s otáčkovým regulátorem měniče SIMOVERT.....	33
6.2	Měřicí část.....	37
6.3	Vyhodnocovací část.....	43

7 ZHODNOCENÍ.....	46
8 ZÁVĚR.....	48
9 SEZNAM POUŽITÝCH ZDROJŮ.....	49

1 ÚVOD

Na začátku je třeba se zamyslet nad důvodem vzniku vlastních simulačních programů, jako jsou např. MATLAB/SIMULINK, Witness, Electronics Workbench, apod.. Jedním z hlavních důvodů byla potřeba zjištění vývojářů, manažerů a jiných různých profesí, jak by se projevila určitá myšlenka, soubor myšlenek v praxi, aniž bychom museli pracovat na reálném systému, při kterém existuje určitá možnost poškození nebo ztráty ať už materiální anebo finanční. Základem kvalitní simulace je ovšem vytvoření modelu, který je, co do vlastností a funkčnosti, do jisté míry shodný s reálným systémem. Je zde možnost přizpůsobení simulace vlivům, které by se na reálném systému jen velmi těžko a dlouze zjišťovaly nebo by nešly zjistit vůbec. Výsledky prováděné simulací jsou poté dále zpracovávány a vyhodnocovány a následně je možné získané poznatky ověřovat a aplikovat v praxi.

Cílem diplomové práce bylo seznámení s frekvenčním měničem SIMOVERT od firmy SIEMENS a programovým vybavením SIMOVIS od stejné firmy, který nám umožňuje daný frekvenční měnič řídit. Dále bylo třeba přiblížit manipulaci s vytvořenými schématy U/f řízení v prostředí MATLAB/SIMULINK a následné zpracování obdržených výsledků. Posledním bodem této práce bylo vytvoření vizualizačního programu v prostředí programu firmy Borland Delphi, který zobrazuje, tím usnadňuje hodnocení dosažených výsledků, data získaná při samotné simulaci a při měření na reálném pohonu.

2 VYSVĚTLENÍ ZÁKLADNÍCH POJMŮ (SYSTÉM, MODEL, MODELOVÁNÍ, SIMULACE)

2.1 Systém

Slovo „systém“ je v dnešní době používáno v mnoha oborech a v mnoha významech. Jeden obor, který často používá simulaci, totiž teorie regulace a technického řízení, vymezuje termín systém dosti přesně (jako objekt se vstupními a výstupními signály svázanými přes své vnitřní stavy pomocí obyčejných diferenciálních nebo diferenčních rovnic), avšak to neznamena, že bychom v simulaci a v modelování chápali systém podobně. V simulaci a modelování se studuje nějaká věc, resp. možné varianty nějaké věci. Abstrakce může nebo nemusí zanedbat význam času. Systém, v němž se od významu času abstrahuje, se nazývá statickým systémem. Systém, jehož čas se nezanedbává a je přitom chápán „newtonovsky“ (dvě „události“ nastaly v systému současně nebo jedna z nich nastala dříve než druhá), se v modelování a simulaci nazývá dynamickým systémem. Simulace se jinými než dynamickými systémy nezabývá. Dynamický systém je v každém okamžiku své existence v jistém stavu. Pro změnu stavu dynamického systému se používá slovo událost. V modelování a simulaci se chápe systém tak, že je složen z prvků. V dynamickém systému se může počet jeho prvků během jeho existence měnit. Prvky systému mají své vlastnosti, které se odborně nazývají atributy.

2.2 Model

Slovo „model“ se používalo v běžné řeči nejprve pro předlohu. V modelování a simulaci je termín model použit pro analogii mezi dvěma systémy (modelovaným a modelujícím). Jsou-li modelovaný i modelující systém statické, říkáme, že daný model je statický model. V simulaci se však uplatní jen tzv. simulační modely, které splňují následující požadavky:

1. Jejich modelující i modelované systémy jsou dynamické systémy.
2. Existuje zobrazení τ existence modelovaného systému do existence modelujícího systému.
3. Mezi stavy σ_1 a σ_2 jsou splněny požadavky na vztahy mezi prvky a jejich atributy.
4. Zobrazení τ je neklesající.

Model je tedy složitá struktura, která váže dva systémy, jejich prvky a jejich atributy, a v případě simulačních modelů i existence obou systémů. Místo termínu „modelovaný systém“ se používá slova originál. V případě, že jde o simulační model, mluví se raději o systému simulovaném a simulujícím, než o modelovaném a modelujícím.

2.3 Modelování

V češtině dostalo slovo modelovat postupem doby několik významů. V oblasti, které se dříve říkalo kybernetika a která se dnes zaplňuje systematickými aplikacemi výpočetní techniky, dominuje anglicky psaná literatura. Avšak jde-li o modelování ve smyslu výzkumné techniky (nebo - jak se často říká - metody poznání), je už obsah tohoto termínu vymezen jasněji, a to v následujícím smyslu:

Podstatou modelování ve smyslu výzkumné techniky je náhrada zkoumaného systému jeho modelem (přesněji: systémem, který jej modeluje), jejímž cílem je získat pomocí pokusů s modelem informaci o původním zkoumaném systému.

V tomto smyslu tedy platí, že vytvoříme model, v němž modelovaným systémem je zkoumaný systém, ale my budeme experimentovat s modelujícím systémem, při čemž cílem bude dozvědět se něco o modelovaném systému.

2.4 Simulace

Simulace je výzkumná technika, jejíž podstatou je náhrada zkoumaného dyn. systému jeho simulátorem s tím, že se simulátorem se experimentuje s cílem získat informace o původním zkoumaném dynamickém systému. V tomto smyslu budeme termín simulace dále chápat. Předně cílem simulace je získat informace o simulovaném systému, zatímco pouhá jeho náhrada simulátorem nestačí. Samotný simulátor nemusí být realizován na číslicovém počítači, ale dnes je takto realizován stále častěji. Zdůrazněme, že aby šlo o simulaci, musí být cílem experimentu se simulátorem snaha dozvědět se něco o simulovaném systému. Některá spojení jako např. statická simulace, která se dříve vyskytla v literatuře, je nejlépe ignorovat, neboť dnes působí paradoxně a tedy neodborně (jako např. hranatá koule); stejně lze doporučit vyhýbat se opačným výrazům: např. dynamická simulace je stejný pleonasmus jako hranatá krychle.

3 FUNKCE FREKVENČNÍHO MĚNIČE SIMOVERT

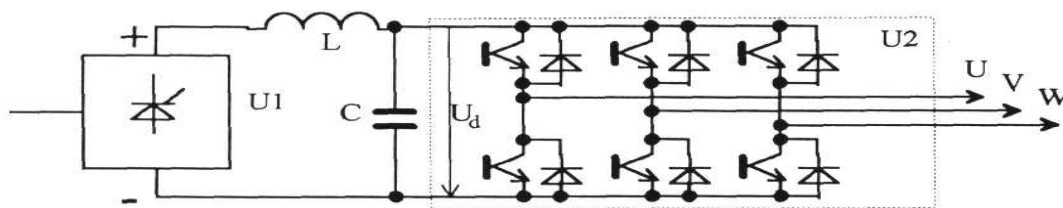
3.1 Měniče pro regulované pohony

Princip frekvenčního měniče spočívá v transformaci vstupního sinusového napětí na napětí výstupní, které má hodnotu amplitudy a frekvence různou od hodnot vstupního sinusového napájecího napětí. Hodnoty frekvence a amplitudy lze nastavovat. Zdroje proměnného kmitočtu při tom musí být schopny dodat AS motoru potřebný činný výkon. Konstrukční uspořádání většinou spočívá na klasické struktuře:

3.1.1 Nepřímé měniče frekvence s napět'ovým meziobvodem:

usměrňovač, LC filtr, střídač. Tento typ je nejužívanějším typem frekvenčních měničů pro AS motory.

Struktura typického měniče s napět'ovým meziobvodem je na obr.3.1.



Obr.3.1 - Měnič s napět'ovým meziobvodem

Měniče umožňují měnit amplitudu výstupního napětí až do hodnoty $U_{JMENOVITÉ}$ se současnou změnou frekvence výstupního napětí.

3.1.2 Přímé měniče frekvence - cyklokonvertory:

skládají se ze šesti třífázových řízených usměrňovačů, které jsou zapojeny antiparalelně. Každý usměrňovač je řízen tak, aby na výstupních svorkách měniče bylo střídavé napětí o potřebné amplitudě a frekvenci. Lze s nimi dosáhnout maximálně 40% napájecí frekvence, proto je tento typ měniče nevhodný pro pomaloběžné pohony s AS motory. Princip činnosti tak spočívá na „chycení správných půlvln napájecí sítě“.

3.2 Základní rysy současných měničů SIMOVERT firmy SIEMENS

Frekvenční měniče SIEMENS Masterdrives (tento typ měniče je užit také v laboratořích TUL) se z hlediska regulační struktury vyrábějí ve dvou modelových řadách a to SIMOVERT VC (Vector Control) a SIMOVERT MC (Motion Control). Modelová řada SIMOVERT VC je určena pro regulaci otáčivé rychlosti AS motorů, řada SIMOVERT MC je určena pro regulaci a polohování třífázových servomotorů. Z hlediska konstrukčního uspořádání se vyrábějí v řadách:

Compact Units - kompaktní měniče s výkony od 2,2 kW do 37 kW,

Chassis Units - měniče ve vestavném provedení s výkony od 37 kW do 200 kW,

Cabinet Units - měniče ve skříňovém provedení s výkony od 45 kW do 1500 kW, a to pro všechny běžné světové napěťové soustavy od 220 do 690V. Z tohoto pohledu je patrné, jakou škálu výkonů pokrývá modelová řada měničů SIEMENS, které se vyrábějí ve dvou provedeních. Buď jako kompletní frekvenční měnič s napěťovým meziobvodem (usměrňovač, napěťový meziobvod a střídač), nebo jen jako napájecí jednotka (třífázový můstek s filtračním kondenzátorem, popř. dva antiparalelně zapojené třífázové můstky).

Měničem lze vytvořit soustavu určenou pro napájení vícemotorového pohonu realizovanou společným stejnosměrným meziobvodem a střídači o různých nebo stejných výkonech. V případě napájení více střídačů ze společného napěťového meziobvodu lze vrácenou energii jednoho střídače využít jiným střídačem. Měničem SIEMENS SIMOVERT lze realizovat pohon s možností jedno-kvadrantového nebo až čtyř-kvadrantového provozu AS motoru s možností rekuperace do sítě nebo brždění do externího odporu (Breaking Unitu).

Celá modelová řada je plně číslicová, základní deska měniče je osazena 32 bitovým RISC mikroprocesorem SIEMENS SAB80165, pracujícím s pohyblivou řádovou čárkou, pamětí RAM a pamětí EEPROM, takže je schopna si zapamatovat nastavení parametrů regulace i po vypnutí napájecího napětí. Doba vzorkování veškerých procesů je od 100μs do 1,5 ms. Výkonové části střídače jsou osazeny tranzistory IGBT a to pro všechna napětí (až do 690 V), což je ve srovnání s jinými světovými výrobci zcela ojedinělé, i z dnešního pohledu. Celá vnitřní struktura měničů je definovaná dle nejmodernějšího přístupu fy SIEMENS tzv. BICO technologii, která umožňuje měnit strukturu měniče dle požadavků na provoz konkrétního zařízení.

Frekvenční měniče řady SIEMENS SIMOVERT zajišťují vnější komunikaci pomocí vestavěného rozhraní případně tzv. technologickou klávesnicí. Samotné parametrizování měniče řady SIMOVERT lze provádět manuálně nebo využitím softwaru SIMOVIS (Drive Monitor), který je možné nainstalovat na běžné PC a umožňuje také vizualizaci základních průběhů a nastavení měniče. Měníče řady SIMOVERT jsou schopny potřebné parametry nastavení pohonu vypočítat zcela autonomně bez nutnosti zásahu obsluhy (při použití AS motoru fy SIEMENS). Celá modelová řada měničů SIMOVERT je vybavena systémem vnitřní ochrany Squards, vyvinutým firmou SIEMENS. Ochrana Squards zajišťuje bezpečný chod pohonu a zároveň ochranu před možným překročením jistých mezí pohonu, jako je např. oteplení a proudová přetížitelnost.

Frekvenční měniče řady SIMOVERT jsou vyráběny v takové škále různých modifikací a provedení, že je nutno pro další práci velmi přesně specifikovat typ a provedení měniče užitého při provádění dalších experimentů. Laboratoře TUL používají AS motor SIEMENS 1LA71066-AA10-2, frekvenční měnič Siemens SIMOVERT 6SE7016-1EA61. Celé zařízení je napojeno na PC s instalovaným softwarem SIMOVIS.



Obr.3.2 - Siemens SIMOVERT MASTERDRIVES Vector Control

Štítkové údaje AS motoru:

Typ:	1LA71066 AA10-2
Výrobce:	SIEMENS
Počet pólů:	6
Napájecí napětí:	Δ / Y 230 / 400V
Frekvence:	50 Hz
Jmenovité otáčky:	1000 ot/min
Jmenovitý výkon:	1,5 kW
Parametry při jmenovitém výkonu:	
Otáčky:	925 ot/min
Účinník – $\cos \varphi$:	0,75
Jmenovitý proud:	Δ / Y 6,8 / 3,9 A
Jmenovitý moment:	15 N*m

Technická specifikace použité jednotky SIMOVERT:

Typ:	SIMOVERT 6SE7016 – 1EA61 – Z	
Jmenovité napájecí napětí:		3*380 V (AC)
Frekvence výstupního napětí:	U/f = konst.	0 – 600 Hz
	U = konst.	8 – 300 Hz
Jmenovitý výstupní proud:		6,7 A
Výkonová ztráta při modulační frekvenci 2,5 kHz:		0,11 k W
Hmotnost:		cca 8,5 kg



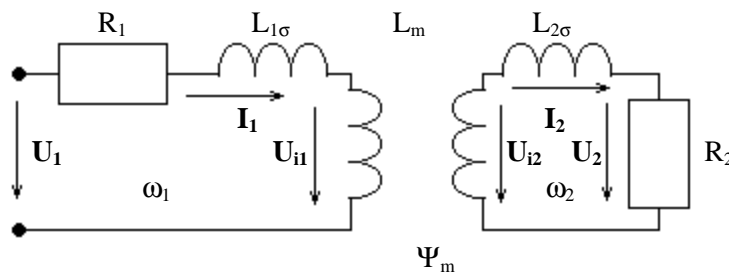
Obr.3.3 – Pohled na reálný systém v laboratořích TUL

4 ŘÍZENÍ ASYNCHRONNÍCH MOTORŮ

Velký rozmach zavádění asynchronních motorů s kotvou nakrátko do většiny pohonů v poslední době způsobilo několik hlavních důvodů:

- nízká cena (levná hromadná či sériová výroba)
- spolehlivost (díky jednoduché konstrukci)
- dobré provozní vlastnosti:
 - mechanická odolnost (motory mohou pracovat v rozličných podmínkách)
 - zátěžová odolnost (krátkodobá proudová a momentová přetížitelnost)
- výhodný poměr výkonu a hmotnosti (velký výkon, nízká hmotnost)

Moment motoru závisí jak na magnetickém toku ve vzduchové mezeře, tak na rychlosti. Poněvadž vztah mezi momentem a magnetickým tokem ani vztah mezi momentem a rychlostí není lineární, je řízení takového motoru velmi složité. Proto se při odvozování vlastností asynchronních motorů s kotvou nakrátko používá náhradní schéma (obr.4.1).



Obr.4.1 - Náhradní schéma AS s oddělenými obvody

Při práci bez náročnějších požadavků na regulaci velikosti momentu a rychlosti se běžně vystačí s řízením v otevřené smyčce (např. pouhé připojení motoru k síti - ventilátory). V případech, kdy jsou požadovány rychlé dynamické změny, je zapotřebí pracovat v uzavřené smyčce. Takové řízení bylo možné až s příchodem rychlé moderní výkonové elektroniky (např. rychlé spínací členy, výkonnější procesory, apod.).

Způsoby řízení asynchronního motoru lze rozdělit v podstatě do dvou hlavních skupin:

- Skalární řízení
- Vektorové řízení

4.1 Skalární řízení

Skalární řízení se používá převážně u takových aplikací, kde není kladen požadavek na vysokou dynamiku pohonů. Při návrhu se vychází z následujících předpokladů:

- řízení se odvozuje z rovnic motoru v ustáleném stavu
- řízením se udržuje konstantní hodnota magnetického toku statoru Ψ_1

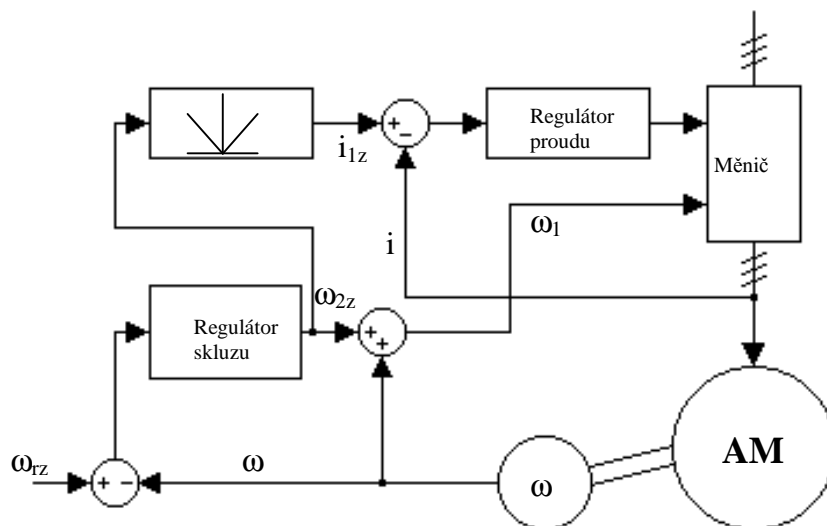
Skalární řízení mohou být různá. Mohou se lišit například ve složitosti regulačního obvodu nebo typech použitých součástí. Podle typu regulace můžeme rozlišovat například regulaci s jednou, dvěma či více zpětnými vazbami. Ve většině regulačních obvodů je v jedné zpětné vazbě zapojen proudový regulátor. Ten na základě získané informace o velikosti statorového proudu reguluje tak, aby nebyla překročena maximální povolená hodnota proudu udávaná výrobcem motoru. Dále se v obvodu může nacházet například regulátor otáček. Podle typu měniče se skalární řízení může dělit na frekvenčně napěťové nebo frekvenčně proudové řízení.

4.1.1 Frekvenčně proudové řízení

Při frekvenčně proudovém řízení se nejprve určí závislost

$$i_1 = f(\omega_2) \quad (4.1)$$

při zachování konstantní hodnoty magnetického toku statoru $\Psi_1 = \text{konst.}$ Tato závislost pro vyšší kmitočty vychází lineární. Při řízení pohonu je statorový kmitočet vytvářen ze součtu naměřené otáčivé rychlosti a zadaného rotorového kmitočtu. Podle hodnoty ω_2 se ze vztahu (4.1) určí žádaná hodnota statorového proudu, kterou regulátor proudu udržuje v meziobvodu frekvenčního měniče. Na obr.4.2 je zobrazeno možné blokové schéma proudového řízení.



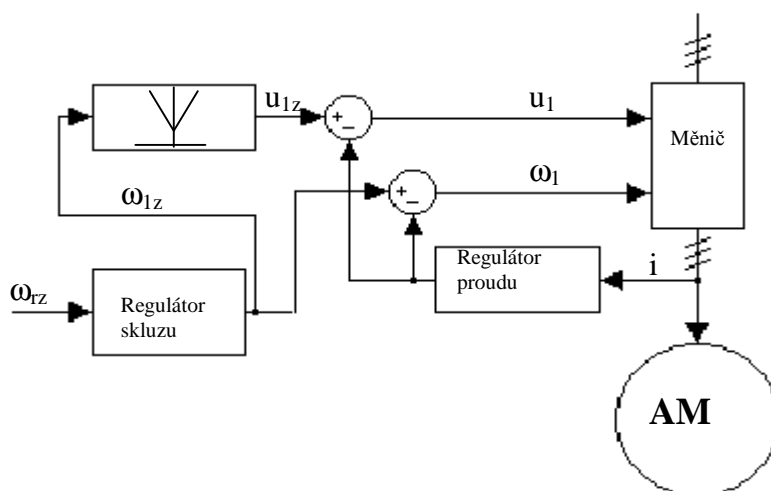
Obr.4.2 - Frekvenčně proudové řízení

4.1.2 Frekvenčně napěťové řízení

Pokud se použije frekvenčně napěťové řízení, určí se nejdříve závislost

$$u_1 = f(\omega_1) \quad (4.2)$$

pro $\Psi_1 = \text{konst.}$ Tato závislost se opět může nahradit lineárním průběhem. Při řízení pohonu statorový kmitočet již není vytvářen ze součtu naměřené otáčivé rychlosti a zadaného rotorového kmitočtu, proto není potřeba tak přesného měření rychlosti jako u řízení frekvenčně proudového. Požadované napětí je vytvořeno jako funkce kmitočtu v závislosti na statorovém proudu.



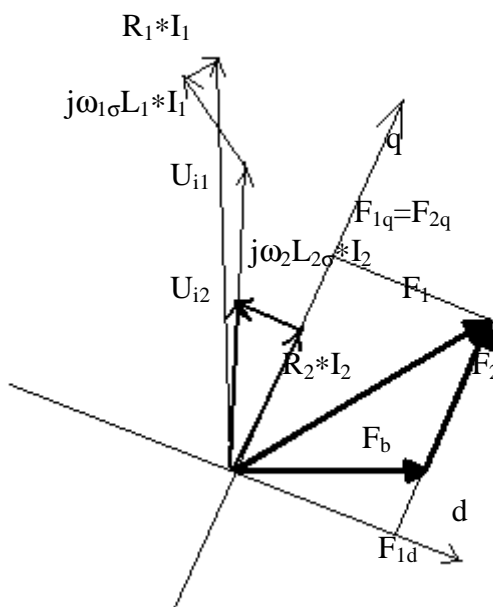
Obr.4.3 - Frekvenčně napěťové řízení

4.2 Vektorové řízení

Již popsané způsoby řízení tedy vycházely ze vztahů pro ustálené stavy. Při dynamicky náročnějších požadavcích tyto metody příliš nevyhovují. Pro řízení těchto pohonů se tedy používá metoda vektorového řízení.

Metody vektorového řízení využívají transformací do pravoúhlých souřadných systémů. Nejčastěji používané jsou transformace do systému α, β (systém pevně spojený se statorem) a do systému d, q (spojený s točivým magnetickým polem motoru). Tyto metody a jejich vlastnosti budou blíže popsány v kapitole 4.3.

Při odvozování principů vektorového řízení se vychází z vlastností již známých z řízení stejnosměrného cize buzeného motoru. U tohoto motoru bylo velké dynamiky dosaženo udržováním konstantního magneto-motorického napětí F_b v budícím vinutí, které je úměrné budicímu proudu i_b a podle potřeby se měnilo magneto-motorické napětí (dále jen MMN) F_a ve vinutí kotvy, které je úměrné proudu kotvy i_a . MMN F_b vytváří magnetický tok stroje, nazývá se tedy tokotvornou složkou, MMN F_a je úměrné momentu motoru, proto se mu říká momentotvorná složka. U cize buzeného stejnosměrného motoru lze tyto složky nezávisle na sobě řídit, protože obě vinutí jsou navzájem oddělena.



Obr.4.4 - Vektorový diagram s rozkladem do os d, q

Protože u asynchronního motoru tyto dva oddělené obvody se dvěma napájecími zdroji pro momentotvornou a tokotvornou složku neexistují, je i realizace vektorového řízení podstatně složitější. Na obr.4.4 je nakreslen příslušný vektorový diagram s vektory MMN \mathbf{F}_1 , \mathbf{F}_2 , \mathbf{F}_b . Při vhodném zvolení os d,q můžeme rozložit vektor \mathbf{F}_1 na složky \mathbf{F}_{1d} a \mathbf{F}_{1q} , kde složka \mathbf{F}_{1d} vytváří v magnetickém obvodu stroje magnetický tok a odpovídá tedy tokotvorné složce. Složka \mathbf{F}_{1q} je rovna složce \mathbf{F}_{2q} a je úměrná úbytku napětí U_2 na rotorovém odporu \mathbf{R}_2 . Složitost vektorového řízení asynchronních motorů spočívá především v určování okamžité polohy vektoru MMN \mathbf{F}_2 vůči statoru. Existují dva druhy metod:

- přímé metody
- nepřímé metody

4.2.1 Přímá metoda vektorového řízení

Metody přímé rekonstruují vektor magnetického toku buď z naměřených hodnot statorových napětí a proudů nebo přímým měřením vektoru magnetického toku ve vzduchové mezeře asynchronního motoru.

Přímé měření vektoru magnetického toku vyžaduje instalaci speciálních čidel. Pro přímé měření polohy magnetického toku ve vzduchové mezeře se používají například Hallovovy sondy nainstalované přímo do asynchronního motoru. K tomu je zapotřebí jiný motor než motor s běžnou konstrukcí vzhledem k velikosti vzduchové mezery. Navíc sondy měří i rušivé signály, jejichž odfiltrování je velice složité, a jsou ovlivněné teplotou. Proto se v poslední době tyto metody příliš nepoužívají.

Při rekonstrukci vektoru magnetického toku z naměřených statorových veličin je potřeba znát i polohu rotoru. Pokud je pro zpětnou otáčkovou vazbu použit estimátor skluzu, lze pro toto řízení použít standardní asynchronní motor bez dodatečně montovaných snímačů polohy nebo rychlosti. Estimátor magnetického toku rotoru je založen na řešení těchto rovnic:

$$\begin{aligned}\Psi_s &= \int (\mathbf{u}_s - \mathbf{i}_s R_s) dt + \Psi_{s0} \\ \Psi_r &= \Psi_s - \sigma L_s \mathbf{i}_s \\ \sin \Theta &= \frac{\Psi_{r\beta}}{|\Psi_r|} \quad \cos \Theta = \frac{\Psi_{r\alpha}}{|\Psi_r|}\end{aligned}$$

4.2.2 Nepřímá metoda vektorového řízení

Nepřímé metody využívají různých způsobů výpočtu polohy magnetického toku. K výpočtům se využívají naměřené hodnoty proudů a napětí na statoru a měření otáček. Tyto veličiny se v regulačním obvodu postupně transformují do různých souřadných systémů. Existuje mnoho modifikací realizace vektorového řízení, ale základní princip rozkladu proudu na dvě složky a jejich oddělené řízení zůstává vždy zachován.

Příklady:

- Nepřímá metoda s čidlem rychlosti rotoru
- Nepřímá metoda s čidly napětí

4.2.2.1 Nepřímá metoda s čidlem rychlosti rotoru

Tento způsob řízení využívá signálu ω z čidla rychlosti na hřídeli motoru a signálu průběhu proudu z čidel ve fázích přívodu ke statorovému vinutí. Proudů se poté transformují do souřadného systému α, β . Z těchto proudů se získá průběh MMN F_α, F_β . Ty se potom transformují do systému d,q, který se otáčí synchronně s magnetickým polem stroje. K tomu potřebujeme znát okamžitou polohu vektoru magnetického pole vůči systému α, β . Tu získáme ze vztahu

$$\Theta_1 = \int d\Theta_1 = \int (\Theta_m + \Theta_2) = \int (\omega + \omega_2) dt$$

Získané signály F_{1d} a F_{1q} jsou skutečnými hodnotami, které jsou využity pro regulaci se žádanými hodnotami F_{1dz} a F_{1qz} . Výstupní signály opět převedené na statorový proud jsou převedeny do systému α, β a poté transformovány do klasického třífázového systému a přivedeny do měniče. Při tomto způsobu řízení je potřeba přesného čidla rychlosti.

4.2.2.2 Nepřímá metoda s čidly napětí

Při tomto řízení se vychází z obdobného zapojení jako při nepřímé metodě s čidlem rychlosti rotoru. Vedle čidel statorového proudu a rychlosti se snímají ještě dvě sdružená napětí U_{UV} a U_{VW} . Okamžitá poloha vektoru magnetického toku se získá rekonstrukcí z vektorového diagramu napětí a magneto-motorických napětí rozložených do os d, q. Zde se využívá úměrnosti mezi úbytkem napětí U_2 na rotorovém odporu R_2 a momentotvornou složkou magneto-motorického napětí F_{1q} .

5 PROGRAM MATLAB/SIMULINK A VYTVOŘENÉ BLOKY

Na TUL je, v učebnách zaměřených na práci s Matlabem, program nainstalován i s potřebnými Toolboxy. Samotný název Matlab je odvozen od zkratk slov Matrix a Laboratory. Z toho je zřejmé, že elementárním datovým objektem je matice. Výpočtové a simulační prostředí Matlab je tvořeno základním výpočtním jádrem, na které je „nabaleno“ mnoho Toolboxů, které lze různě instalovat, aktualizovat a tím rozšiřovat možnosti Matlabu. Všechny operace v Matlabu jsou založeny na aplikaci speciálního prostředku S-funkce. S-funkce může být ve formě grafické nebo programové. Grafická forma je užívána např. v Simulinku, zatímco programová je použita v souborech **.m* nebo **.mex*.

Největší oblibě se však těší právě již zmiňovaný Toolbox Simulink.

Samotný Simulink se skládá z několika předem definovaných knihoven, jako jsou knihovny:



Sources - zdroje signálů



Sinks - zviditelnění práce se signály např.: osciloskop, display



Discrete - diskrétní prvky



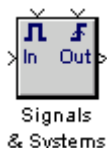
Continuous - lineární prvky



Nonlinear - nelineární prvky např. přepínače, sycení, hystereze



Math - matematicky definovatelné bloky např. funkce, zesílení



Signal & Systems - bloky pro práci se signály, multiplikaci definování subsystémů.

Bloky předdefinované v Simulinku jsou základním prvkem pro tvorbu schémat. Schémata jsou sestavena pouhým pospojováním bloků a simulace běží již ve výpočetním jádru Matlabu.

Pro simulaci elektrických pohonů byl vyvinut Toolbox - Power System Blockset, který obsahuje celou škálu prvků aplikovatelných v elektrotechnice a výkonové elektronice.

Power System Blockset se skládá ze skupin bloků:

Connectors - propojující prvky

Electrical Sources - obsahuje celou škálu elektrických zdrojů a to jak stejnosměrných tak i střídavých

Elements - obsahuje prvky užívané v elektrotechnice jako cívky, vypínače, RLC členy.

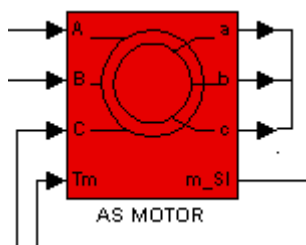
Machines - blok zahrnující elektrické stroje a motory různých typů a provedení

Measurements - blok obsahující voltmetr a ampérmetr

Power Electronics - blok výkonové elektroniky zahrnující bloky tyristor, dioda, tranzistor, tranzistory mosfet atd..

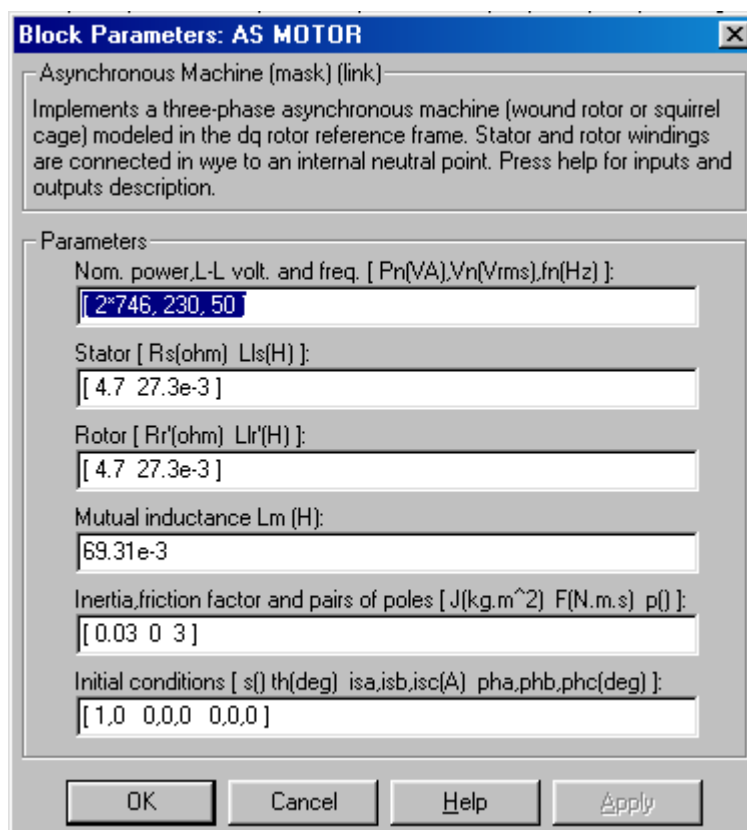
Samostatně stojící blok Powergui je blok sledující činnost celých elektrotechnických schémat a zobrazující základní informace o prováděných simulacích.

Tato studie je založena na bloku AS Motoru



Obr.5.1 - Blok AS MOTOR

Blok AS MOTOR umožňuje měnit parametry, které lze snadno zadat do dialogového okna, otevírajícího se při „dvojkliku“ na blok AS MOTOR (dialogové okno viz obr.5.2), samostatně lze zadat na vstupu bloku AS MOTOR zatěžující moment $T_m[N]$.



Obr.5.2 - Dialogové okno bloku AS MOTOR

Vstupním signálem pro blok AS MOTOR musí být jako u skutečného AS motoru napětí, které je získáno převedením běžného signálu používaného v Simulinku pomocí bloku Controlled Voltage Source. Na vstup bloku je zde přiveden signál, který se má změnit na napětí, jeden z výstupů bloku musí být uzemněn a druhý výstup představuje transformovanou hodnotu signálu na napětí. Průběh napětí je, co do velikosti a tvaru signálu, zcela identický se vstupním signálem. Z toho je patrné, že blok AS MOTOR nelze napájet běžným signálem, užitým v klasických aplikacích Simulinku.

Dalším problémem aplikování Power System Blocksetu je nutnost použít alespoň jeden blok Voltage Measurement, který musí být nutně použit ve schématu tvořeném za pomoci Power System Blocksetu. Tento blok funguje jako běžný voltmetr a také se s ním tak pracuje, to znamená, že na vstupy je přiveden různý potenciál napětí a jeho

hodnota se na výstupu zobrazuje na připojeném bloku osciloskop, případně i na bloku display. Obdobně jako blok Voltage Measurement se chová blok Current Measurement. Ten představuje blok ampérmetru a podle toho se s ním tak nakládá, na výstupu bloku je pak hodnota proudu.

Schémata využívající blok AS MOTOR musí dále obsahovat blok Asynchronous Machine Demux, který je součástí Power System Blocksetu. Tímto nástrojem lze sledovat až 20 parametrů z výstupu AS MOTORU najednou (výstup AS motoru je označen m_SI):

Výstupy **1-3**: rotorové proudy i'_{ra} , i'_{rb} , i'_r

4-9: i'_{qr} , i'_{dr} , ϕ'_{qr} , ϕ'_{dr} , v'_{qr} , v'_{dr}

10-12: statorový proud i_{sa} , i_{sb} , i_{sc}

13-18: i_{qs} , i_{ds} , ϕ_{qs} , ϕ_{ds} , v_{qs} , v_{ds} ;

19-20: ω_r , T_e

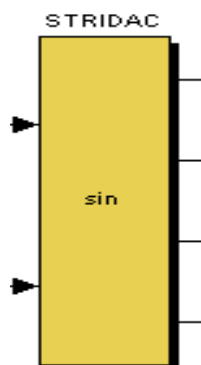
S těmito veličinami lze dále již pracovat jako se signály, přestože se jedná o napětí, případně proud.

5.1 Modelování skalárního řízení měniče SIMOVERT

K modelování frekvenčního měniče SIEMENS SIMOVERT bylo použito prostředí Matlab/Simulink společně s Power System Blocksetem.

Základní úlohou při simulaci U/f řízení v Matlab Simulinku bylo vytvořit model střídače, který by fungoval jako obvod střídače u frekvenčního měniče s napět'ovým meziobvodem a vystihoval také jeho dynamické chování.

Při tvorbě modelu střídače bylo nutno vytvořit samostatný blok v Simulinku, který by vytvářel napětí o žádané hodnotě amplitudy a frekvence výstupního 3-fázového napětí.

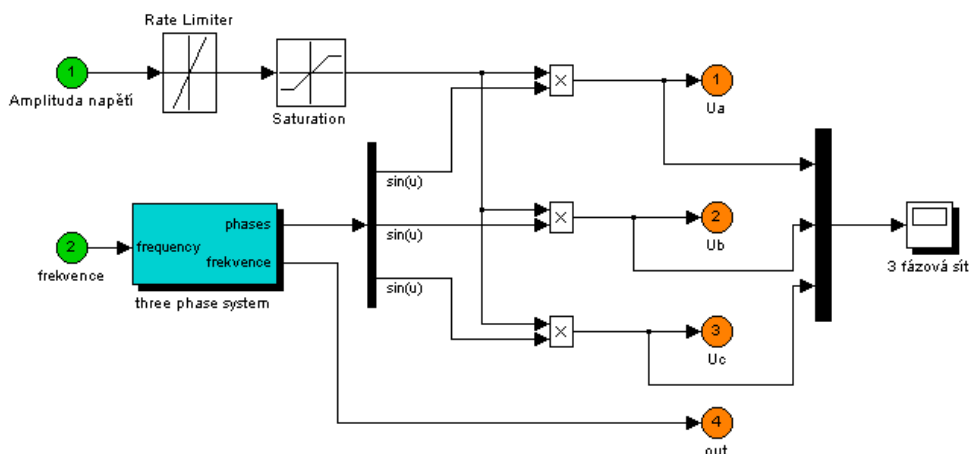


Obr.5.3 - Blok STŘÍDAČ

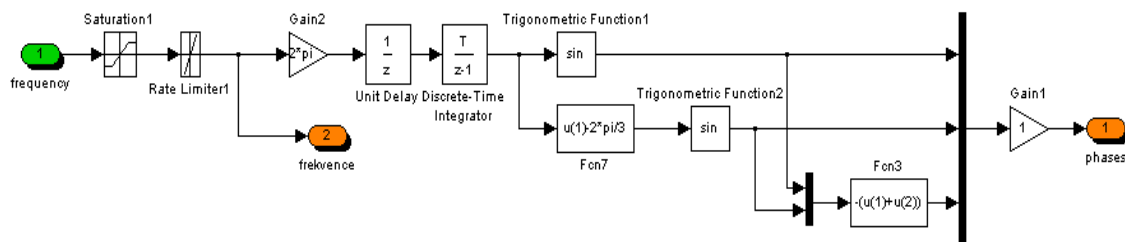
Pro vytvoření bloku střídače existují dvě základní cesty:

- 1) využití PWM – pulsně šířkové modulace, která by byla vygenerována prvky Simulinku a následně vygenerovanou spínací sekvencí užít ke spínání výkonových prvků (tyristorů), které obsahuje knihovna Power System Toolbox. Experimentování s touto myšlenkou se však ukázalo jako zcela zavádějící, protože prováděné simulace probíhali velmi nereálně a pomalu. Matlab/Simulink se ukázal pro tuto cestu takřka nepoužitelný.
- 2) využít základních bloků Simulinku, napětí o potřebné frekvenci a amplitudě vygenerovat pomocí bloků Trigonometric Function, konkrétně sinusová funkce. Do bloku STŘÍDAČE vstupuje pak na sobě zcela nezávisle amplituda napětí, frekvence a vystupuje trojfázová soustava s parametry žádané frekvence a amplitudy 3-fázového napětí. Z bloku STŘÍDAČE dále vystupuje frekvence napětí aktuální trojfázové sítě. Blok STŘÍDAČE (obr.5.3) sestavený z těchto základních bloků byl pak odzkoušen a použit v dalších aplikacích. Zjednodušující podmínkou je pouze fakt, že spínací frekvence výkonových prvků je řádově několikrát vyšší než frekvence výstupní 3-fázové napájecí sítě a vlastní spínací frekvence ji nijak neovlivňuje, zanedbává se PWM modulace. Změny amplitud napětí jsou tedy téměř okamžité.

Realizace bloku je na obr.5.4, ze kterého je patrná tvorba sinusového signálu s amplitudou 1 o požadované hodnotě frekvence, posunuté navzájem mezi sebou o 120° . Na obr.5.5 je patrná již samotná tvorba 3-fázového napětí o žádané frekvenci a amplitudě.



Obr.5.4 - Vnitřní uspořádání bloku STŘÍDAČE



Obr.5.5 - Vnitřní uspořádání bloku třífázového systému

STŘÍDAČ, vytvořený pomocí bloků Trigonometric Function, je ve své struktuře na vstupech doplněn o bloky saturací, které zabraňují tomu, aby se na vstupy žádaných hodnot amplitudy napětí a frekvence dostaly do střídače nereálné hodnoty nebo hodnoty záporné (mohlo by tak nastat při některých regulačních pochodech). Vstup STŘÍDAČE je dále doplněn o bloky Rate Limiterů, které vyhlazují prudké dynamické změny, hodnoty horních a dolních limitních mezí byly určeny experimentálně a to tak, aby bylo dosaženo maximálních možných dynamických změn na vstupu bez nepříznivých vlivů na výstupní trojfázovou síť a byla zachována podmínka zjednodušení.

5.2 Blok U/f CHARAKTERISTIKA

Pro skalárně řízený pohon s měničem Siemens SIMOVERT je základním řídicím kanálem žádaná frekvence f , které se přiřazuje hodnota napětí U .

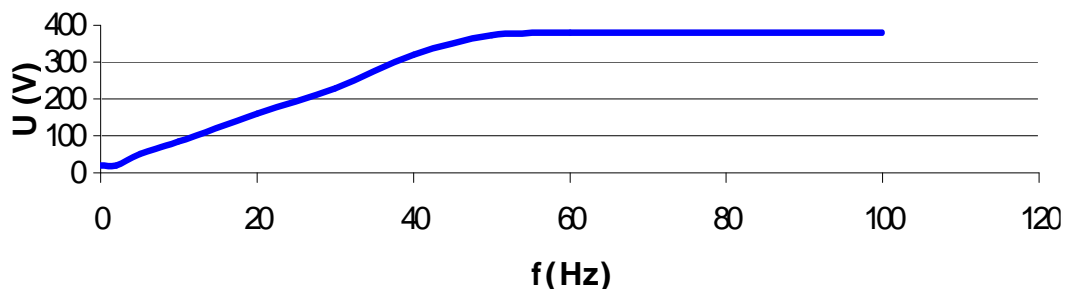
U/f charakteristika může být v měniči SIMOVERT lineární nebo kvadratická, případně ji lze celou nadefinovat pomocí parametrů určených např. prostředím SIMOVIS. U/f charakteristika je měřitelná ze stávajících defaultně nadefinovaných hodnot měření na měniči frekvence.

Této možnosti bylo využito i při realizaci bloku U/f CHARAKTERISTIKY v prostředí Matlab/Simulink, tab.5.1.

Tab.5.1

NAMĚŘENÁ U/f CHARAKTERISTIKA										
FREKVENCE f [Hz]	0	2	5	10	20	30	40	50	60	100
NAPĚTÍ U [V]	20	20	51	84	160	230	320	374	380	380

U/f CHARAKTRISTIKA



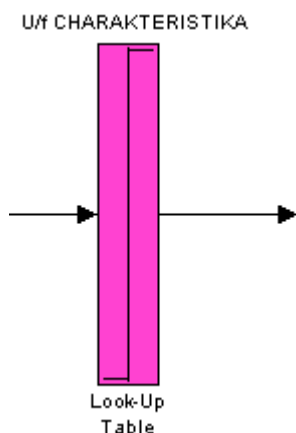
Obr.5.6 U/f charakteristika měniče SIMOVERT

Defaultně nastavená hodnota u měniče Siemens SIMOVERT má lineární průběh napětí v závislosti na frekvenci výstupního napětí měniče. Takovému nastavení odpovídá hodnota parametru měniče $P330=0$ (což je lineární charakteristika) a $P325=20V$.

Blok U/f CHARAKTERISTIKA je realizován pouhým využitím bloku Simulinku Look-Up Table, který je definován vektory x a y . Vektory x a y jsou určeny pomocí souboru **.m* určeného k parametrizování příslušných schémat. Jedná se tedy o soubory: *sch_.m*, *sch_tex.m*, *sch_irc.m*.

Předefinování bloku U/f CHARAKTERISTIKA je tedy umožněno za pomoci pouhého přepsání příslušných vektorů v určeném souboru.

Výsledný blok U/f CHARAKTERISTIKA, který byl použit při simulaci skalárního řízení je na obr.5.7.



Obr.5.7 - Blok U/f CHARAKTERISTIKA

6 NÁVRH LABORATORNÍHO CVIČENÍ

Úkol:

Úkol bude rozdělen do 3 částí:

- 1) Simulační část – seznámení a práce se simulačními schématy Matlab/Simulinku a výsledkem simulací by měly být textové soubory s daty.
- 2) Měřicí část – seznámení a práce s programem SIMOVIS a měření na reálném pohonu, výsledkem měření budou opět textové soubory s daty.
- 3) Vyhodnocování části 1 a 2 – seznámení a práce s vizualizačním programem VizData.

6.1 Simulační část

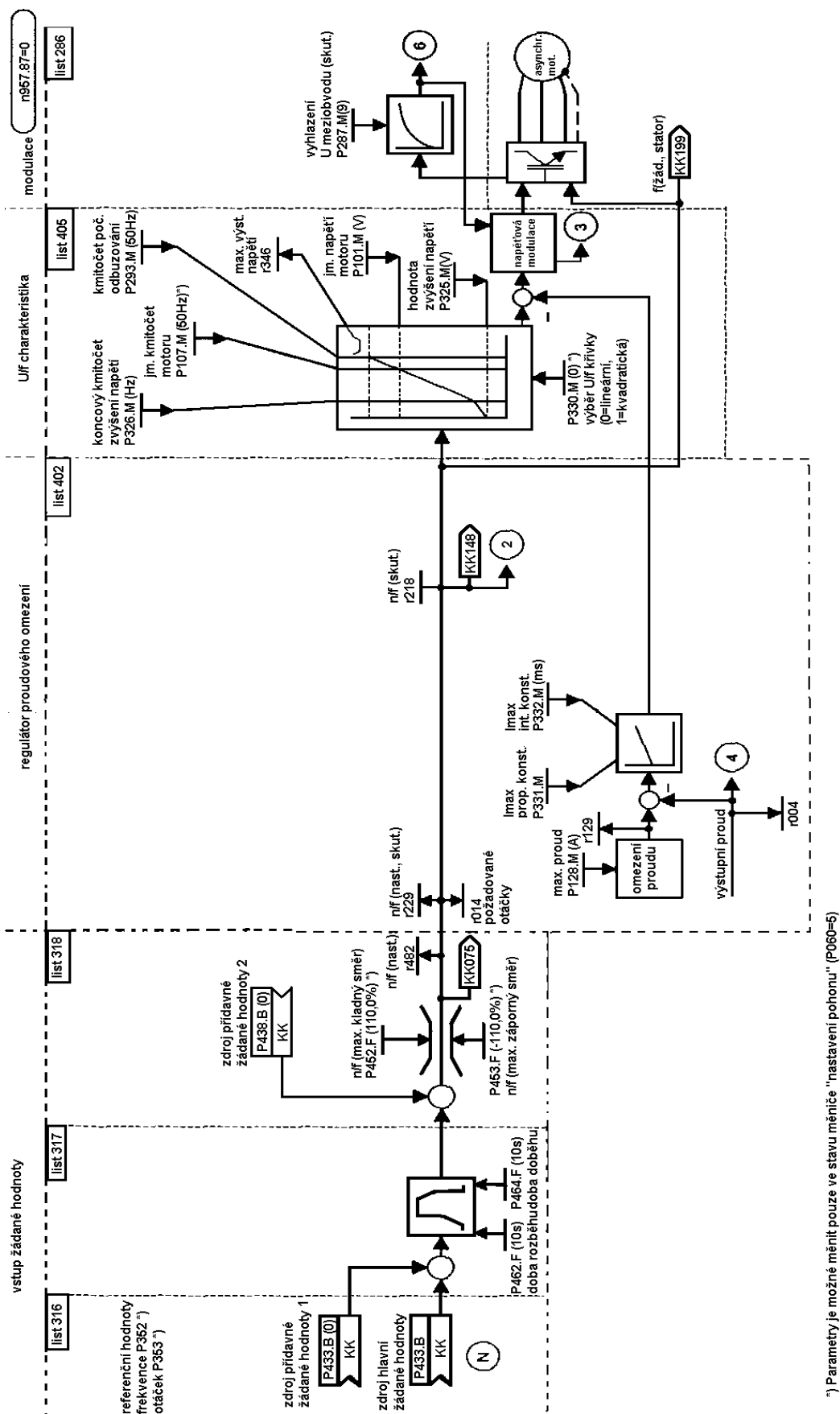
V této části cvičení se zaměříme na chování modelů U/f charakteristiky pohonu v Matlab/Simulinku. Máme k dispozici 3 schémata zapojení:

- 1) Model U/f řízení pro textilní aplikace měniče SIMOVERT
- 2) Model U/f řízení měniče SIMOVERT
- 3) Model U/f řízení s otáčkovým regulátorem měniče SIMOVERT

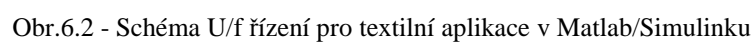
Každé z těchto simulačních schémat si popíšeme, abychom věděli na jakém principu fungují a jak s nimi provádět simulování. Všechny simulační soubory jsou součástí přiloženého CD.

6.1.1 Model U/f řízení pro textilní aplikace měniče SIMOVERT

Model U/f řízení je sestaven na základě parametrizačních schémat frekvenčního měniče Siemens SIMOVERT a to konkrétně na základě schématu, které odpovídá obr.6.1. Ze schématu vychází separabilita systému a je jím určena rozlišovací úroveň pro tvorbu modelu. Rozlišovací úroveň je omezena přístupem k podrobnějším schémátům, některé dílčí informace lze nalézt v Compendiu k měniči SIMOVERT. Analýzou schématu bylo odvozeno základní regulační schéma zobrazené na obr.6.2. Toto schéma najdeme pod souborem simulinku *Uf_tex.mdl*. Dále budeme potřebovat soubory *Sch_tex.m* a *Uloz_data_tex.m*. Tyto 3 soubory musí být uloženy v jednom adresáři.



Obr.6.1 – U/f charakteristika pro textilní aplikace (P100=2)



Princip funkce systému - U/f řízení pro textilní aplikace je dán tímto způsobem: Na vstup žádané hodnoty přichází hodnota žádaných otáček, kterými se má AS motor otáčet. Vzhledem k tomu, že se jedná o pohon se skalárním řízením, jehož princip funguje na základě toho, že příslušné frekvenci je přiřazena hodnota napětí, provede se transformace otáček na příslušnou frekvenci, která náleží výsledným otáčkám pohonu. Hodnota frekvence prochází hlavní větví do bloku, který omezuje maximální možné nárůsty frekvence nebo otáček za časovou jednotku (tento blok je v simulačním schématu zakomponován do bloku STŘÍDAČ v podobě bloků Rate Limiter). Frekvence je ovlivňována proudovým regulátorem, který frekvenci vstupující do hlavní větve může snižovat v případě, že proud tekoucí do AS motoru překročí mez maximálního proudu (= 6,8 A, nastavenou manuálně). Frekvence dále vstupuje nezávisle na sobě:

- 1) do bloku U/f CHARAKTERISTIKA, kterým se přiřazuje hodnota napětí příslušející dané frekvenci. Napětí určené U/f charakteristikou je přesně nastaveno napět'ovým meziobvodem střídače.
- 2) do bloku STŘÍDAČ, který již má připravenou hodnotu napětí napět'ovým meziobvodem. Napětí je pak dále PWM modulací upravené tak, aby výstupní hodnota z bloku střídače měla amplitudu napětí shodnou s napětím připraveným napět'ovým meziobvodem a frekvencí určenou hlavní větví střídače.

Tento typ skalárního řízení je typ s tzv. otevřenou smyčkou a hodí se především pro pohony u nichž není kladen důraz na vysokou přesnost a dynamiku, případnou chybu mezi žádanými a skutečnými otáčkami lze zanedbat.

Proudový regulátor je typu PI – proporcionálně integrační. Regulátor zpracovává regulační odchylku tak, že účinky složek PI regulátoru jsou následující:

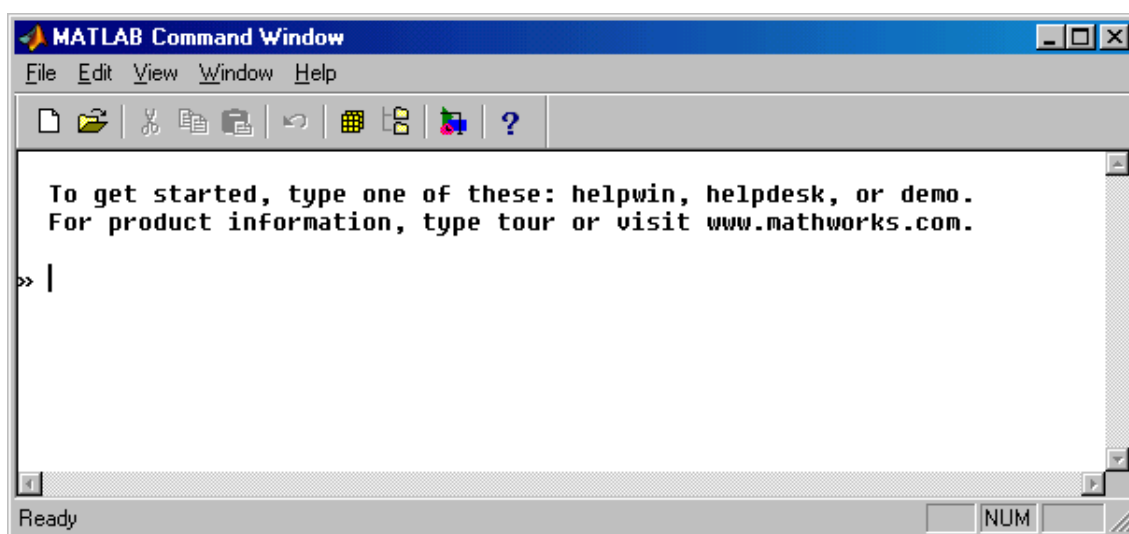
- Proporcionální složka regulátoru reaguje ihned na regulační odchylku na vstupu regulátoru, je úměrná regulační odchylce a v případě trvalé regulační odchylky tuto následně zvyšuje. Regulace proporcionální složkou je statická.
- Trvalou regulační odchylku odstraňuje integrační složka regulátoru, účinek integrační složky se projevuje pomaleji než proporcionální a regulace integrační složkou je astatická. Vhodným spojením obou složek se nevýhody obou složek regulátoru potlačují a projevují se naopak výhody.

Samotný uzavřený regulační obvod proudového regulátoru v U/f řízení pro textilní aplikace je viditelný ve schématu na obr.6.1. Z tohoto vyplývá, že akční veličinou u regulátoru je napětí, hodnotu požadovaného maximálního proudu je zadána manuálně, regulovaná hodnota y je skutečný proud odebíraný AS motorem. Proudový


regulační obvod je zpětnovazební regulační obvod, realizovaný PI regulátorem, který funguje jako proudový omezovač, ochraňující AS motor před proudovým přetížením. Regulační struktura skalárního řízení pro textilní aplikace s proudovým regulátorem realizovaným výše popsaným způsobem je nejjednodušší regulační strukturou skalárního řízení u frekvenčního měniče Siemens SIMOVERT, a proto byla také realizována jako první. Simulace takto popsané regulační struktury v Matlab/Simulinku je zobrazena na obr.6.2.

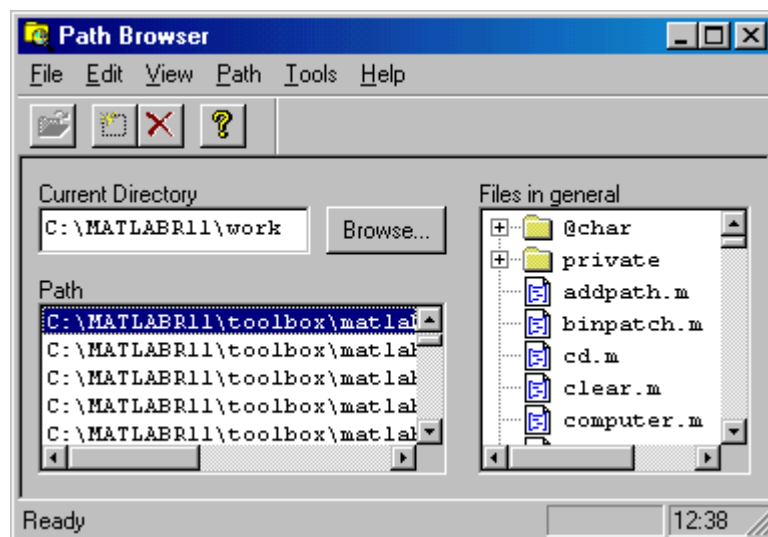
Nyní si popíšeme jak budeme s těmito soubory nakládat, na jakém principu pracuje dané schéma, a jak se dobereme ke kýženému výsledku totiž souboru s daty. V několika krocích bude nastíněn postup (je založen na zkušenostech a znalostech autora).

1) Nejprve si musíme spustit program MATLAB. Otevře se dialogové okno, přesněji „MATLAB Command Window“ (obr.6.3).



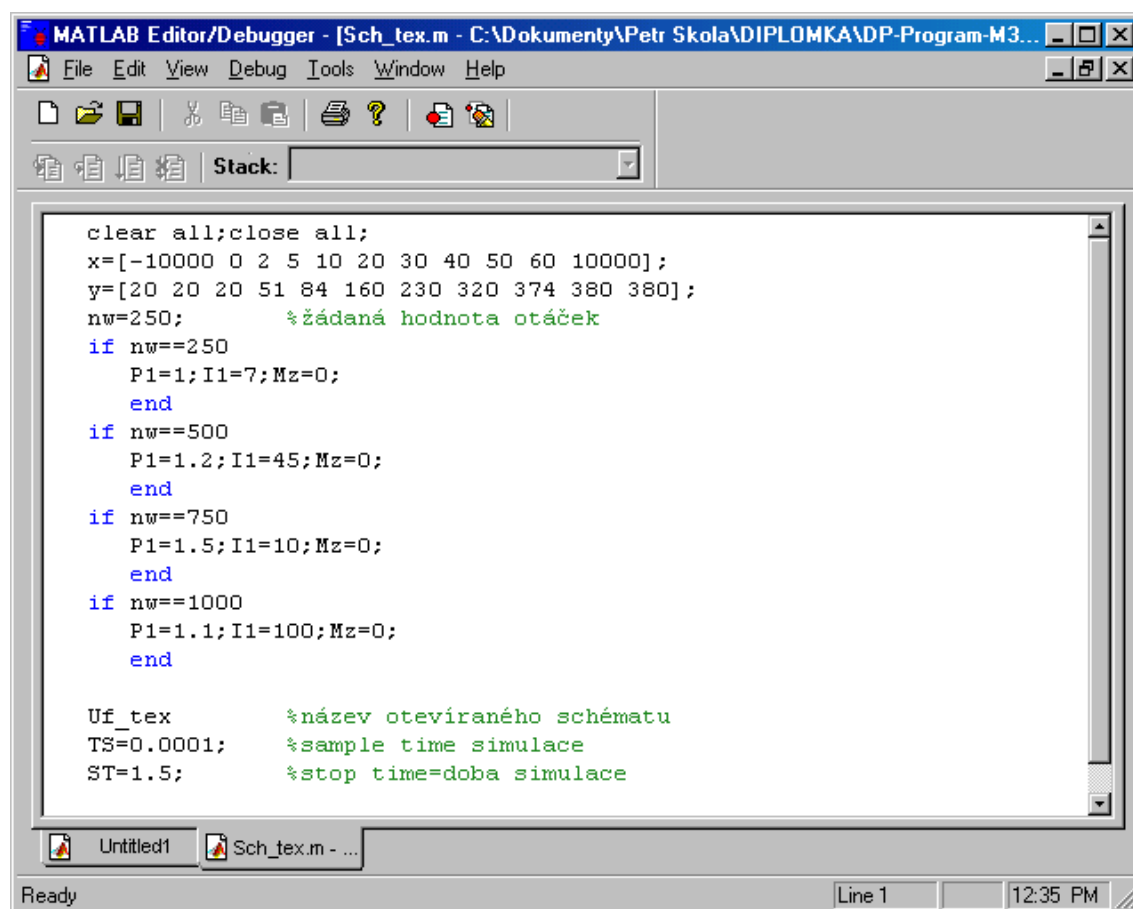
Obr.6.3 - Dialogové okno matlabu

2) Dále je třeba nastavit cestu k našemu souboru, takže musíme spustit „Path Browser“ a to ikonkou  na panelu dialogového okna. Ve kterém pomocí tlačítka „Browse...“ (viz obr.6.4) nastavíme cestu k našemu souboru. O správném nastavení cesty se přesvědčíme v hlavním okně Matlabu a sice příkazem **>>dir**. Měly by se nám vypsat všechny soubory v našem adresáři.



Obr.6.4 - Okno Path Browser

3) Dále si spustíme „MATLAB Editor/Debugger“ a to tak, že v okně „Path Browser“ ⇒ na záložku View ⇒ položka Editor/Debugger. Zobrazí se okno „MATLAB Editor/Debugger“.



Obr.6.5 - Okno Editor/Debugger – soubor Sch_tex.m

Pokud jsme učinili krok 2. správně, tak se nám po kliknutí na ikonu otevření souboru objeví obsah našeho adresáře s uvedeným souborem. Otevřeme tedy soubor **Sch_tex.m**. V souboru (viz obr.6.5) je napsán program, jež nám umožňuje nastavit a měnit parametry měniče a regulátoru potřebné pro běh simulace:


nw – požadované otáčky; **P1** – proporcionální složka PI regulátoru – proudového omezovače; **I1** – integrační složka PI regulátoru – proudového omezovače; **Mz** – zátěžný moment AS motoru; **TS** – perioda vzorkování; **ST** – stop time = doba simulace

Dále jsou zde nadefinovány vektory **x** a **y** pro určení U/f charakteristiky (více kap.5.2) potom se zde vyskytuje název **Uf_tex**, což je název otevíraného schématu nám již známého z obr.6.2.

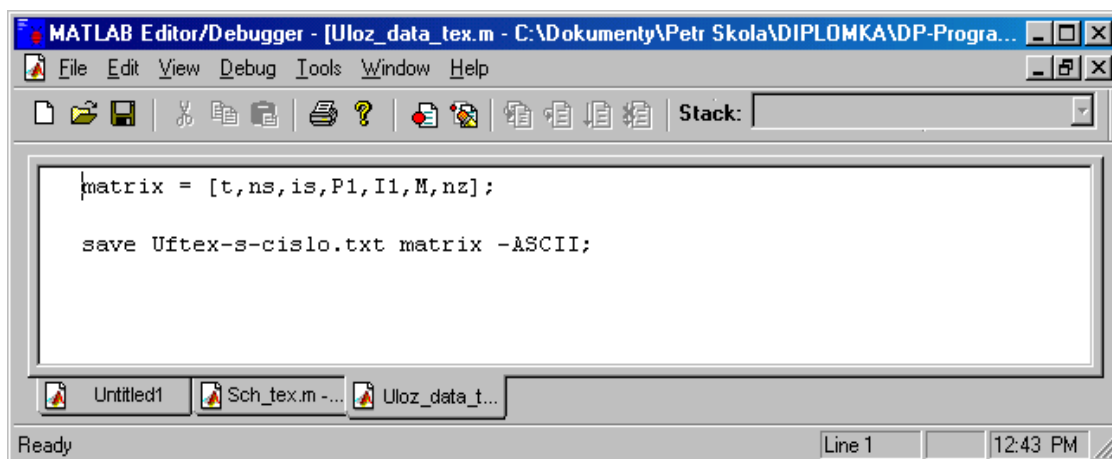
4) Po spuštění – v menu **Tools** \Rightarrow položka **Run**, se tyto parametry zapíše do spouštěného schématu vytvořeném v Simulinku. Při každém novém spuštění souboru **Sch_tex.m** se hodnoty ve schématu přepíše. Změna žádaných otáček **n_w** se děje tak, že v souboru **Sch_tex** se přepíše na 4. řádku **nw=...** hodnota otáček, nejsou-li zadane otáčky předdefinované je třeba upravit program následujícím způsobem:

```
přidáme 3 řádky: if nw==..požadované otáčky
                  P1=..hodnota;I1=..hodnota;Mz=..hodnota;
                  End
```

a následně se provede uložení a spuštění souboru příkazem **Run**. V případě „nepěkných“ průběhů začne ladění některých parametrů a sice změnou **P1**, **I1** a **Mz** přepsáním v souboru nebo nastavením přímo ve schématu **Uf_tex**, pak je ale potřeba ošetřit vstupy do bloků, které jsou nastaveny parametricky (osobně se přimlouvám za první variantu – přepis v souboru). Hodnota žádaných otáček **nw** by se však musela zadat do M-filu.

5) Spustíme simulační schéma **Uf_tex**. To provedeme buď kliknutím na ikonu  nebo v Menu \Rightarrow Simulation \Rightarrow Start a nebo stiskem kláves CTRL+S. Průběh simulace vidíme v dolní části okna. Výsledky si je možné prohlédnout v blocích Scope příslušných veličin.

6) Nyní si potřebujeme naměřená data uložit. Proto je regulační schéma sestavené v Matlab/Simulinku doplněno o bloky To Workspace, což je blok Simulinku zajišťující průběžné ukládání hodnot na vstupu bloku do Workspacu = pracovního prostoru Matlabu. V okně Editor/Debugger si otevřeme soubor **Uloz_data_tex.m** (obr.6.6).



Obr.6.6 - Okno Editor/Debugger – soubor Uloz_data_tex.m

Po spuštění programu je pracovního prostoru dále utvořena matice dat (**t**-čas, **ns**-otáčky skutečné, **is**-proud skutečný, **P1**-proporcionální složka proudového regulátoru, **I1**-integrační složka proudového regulátoru, **M**-moment zátěžný, **nz**-otáčky žádané) uložená do souboru *Uftex-s-cislo.txt*, kde:

- část názvu „*Uftex*“ nám říká, že jde o data získaná ze schématu pro textilní aplikace
- část „*s*“, že jde o data získaná simulací
- část „*cislo*“ – toto označení je jen dočasné, a to z důvodu: spustíme-li vše od kroku 3 pro jiné parametry tento soubor se nám přepíše, proto musíme ihned po uložení část „*cislo*“ nahradit pořadovým číslem měření (1, 2, 3,...,100).

POZOR!!! Musím předeslat, že tato část souboru je velmi důležitá a musí korespondovat s číslem souboru měření. Tzn. Mám-li pod názvem *Uftex-s-1.txt* uloženy data pro žádané otáčky 1000 ot/min, pak v souboru *Uftex-m-1.txt* musím mít uloženy data pro otáčky 100%. Jinak v programu VizData nebudou porovnávána odpovídající si data (viz kap.6.3.).

Vytvořený datový soubor slouží k vizualizaci průběhu otáček a proudu v programu VizData.

6.1.2 Model U/f řízení měniče SIMOVERT

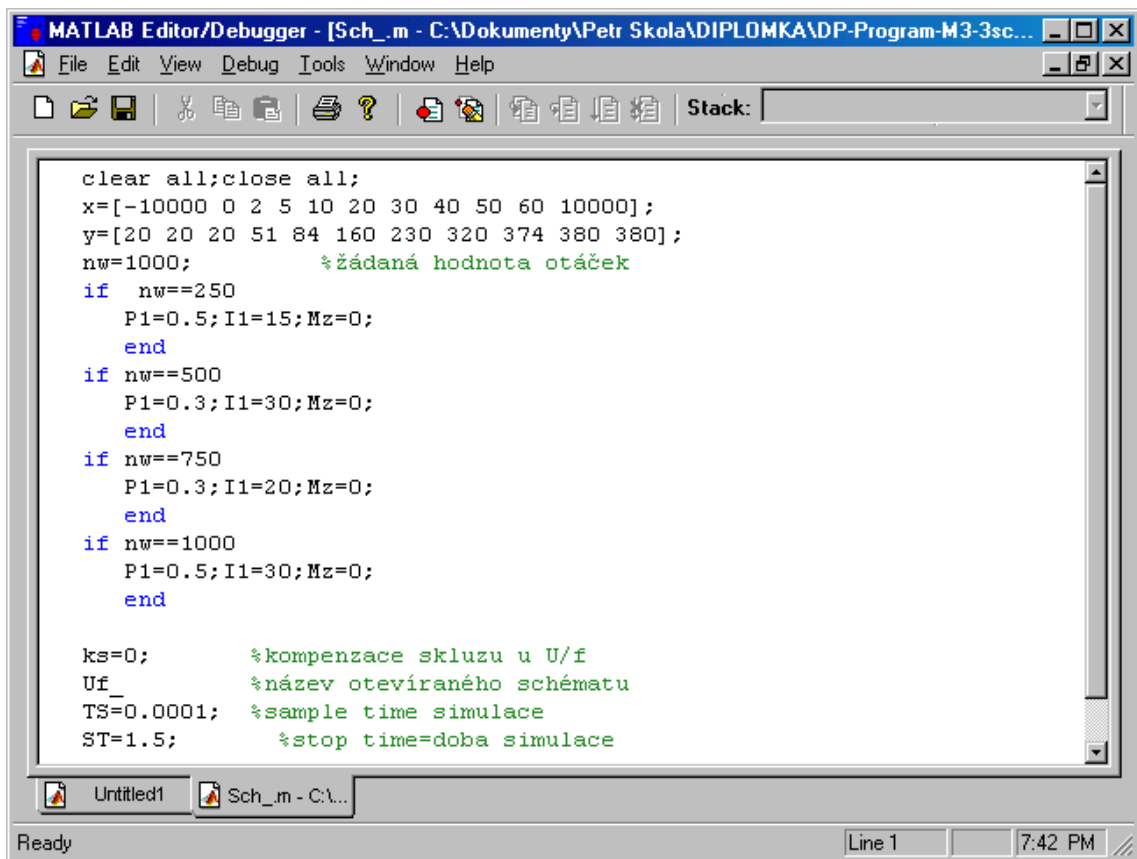
Obdobně jako model U/f charakteristiky pro textilní aplikace měniče SIMOVERT vychází model U/f charakteristiky z parametrizačního schématu U/f charakteristiky měniče frekvence Siemens SIMOVERT. Výchozí schéma zobrazuje obr.6.9. Ze schématu vychází separabilita systému a je jím určena rozlišovací úroveň pro tvorbu modelu obr.6.10.

Princip systému U/f řízení je v podstatě shodný s U/f řízením pro textilní aplikace s tím rozdílem, že proudový regulátor zasahuje do regulačního obvodu ve dvou místech. Akční veličinou regulátoru je pak kromě frekvence zasahující na vstupu do bloku U/f charakteristiky také napětí, které zasahuje do výstupní hodnoty napětí z U/f charakteristiky v případě, že hodnota proudu je vyšší než hodnota proudového omezení ($= 6,8 \text{ A}$). Proudový regulační obvod je tedy opět regulační obvod, realizovaný PI regulátorem, fungujícím jako proudový omezovač, ochraňující AS motor před proudovým přetížením. Realizování proudového regulátoru popsáním způsobem zvyšuje rychlost se kterou je dosažen nebo omezen proud tekoucí do AS motoru z výstupu měniče. V porovnání s regulací U/f pro textilní aplikace je U/f řízení zaměřeno více na dosažení žádaných otáček n_w . K větší rychlosti změny proudu tekoucího do AS motoru je využita velká citlivost AS motoru na změny napájecího napětí, které je dosaženo daleko razantnějším způsobem než v případě U/f řízení pro textilní aplikace. Stejně jako v případě U/f řízení pro textilní aplikace se jedná o uzavřenou zpětnovazební regulační smyčku. Typ řízení U/f je vhodný pro dynamicky méně náročné typy pohonů, u kterých není na závalu trvalá odchylka skutečných otáček od otáček požadovaných na vstupu hlavní žádané veličiny. U tohoto typu řízení může být požadována větší přesnost proudového řízení. Příznivě na regulační pochod působí vyrovnaní skluzu AS motoru, který vznikne zatížením AS motoru konstantním momentem. Parametr kompenzace skluzu zvyšuje hodnotu vstupní frekvence o frekvenci kompenzační, zvyšuje jakost samotné regulace otáček pohonu. Skutečná velikost odchylky je ovlivněna především přesností určení kompenzace skluzu. Simulační schéma popsané regulační struktury v Matlab/Simulinku je zobrazeno na obr.6.10.

Toto schéma najdeme pod souborem simulinku *Uf.mdl* a je vytvořeno z obdobných prvků jako v případě U/f charakteristiky pro textilní aplikace. Dále budeme potřebovat soubory *Sch.m* a *Uloz_data.m*. Tyto 3 soubory musí být uloženy v jednom adresáři stejně jako u předchozího případu.

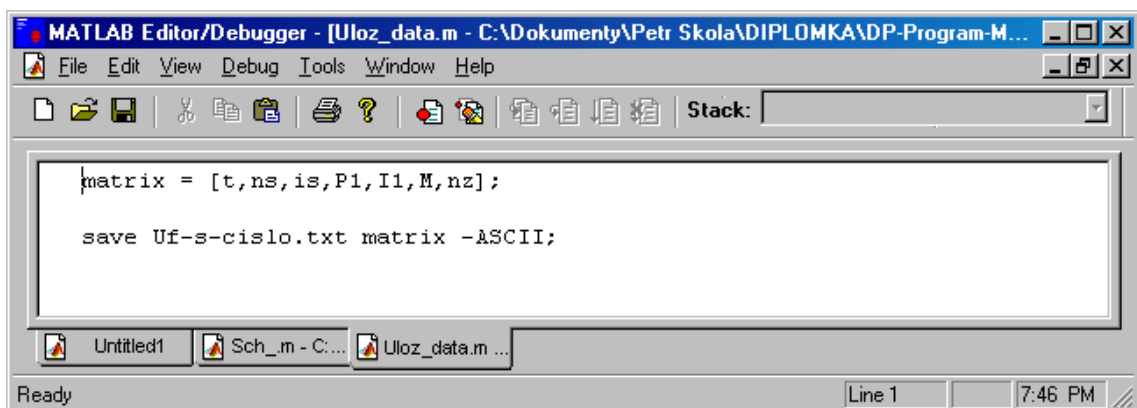
Při spouštění a následné simulaci postupujeme ve stejných krocích jako v kap.6.1.1 s tím rozdílem, že:

- v kroku 3. otevřeme soubor *Sch.m*.
- dále pak v otevřeném souboru můžeme kromě již zmíněných parametrů zadat také parametr k_s – což je kompenzace skluzu (obr.6.7)



Obr.6.7 - Okno Editor/Debugger – soubor Sch_.m

- v kroku 4.se nám po spuštění souboru otevře simulační schéma **Uf_.mdl** (obr.6.10)
 - po spuštění a dokončení simulace následuje krok 6., zde si ovšem, oproti minulému, otevřeme soubor **Uloz_data_.m**, po spuštění se nám data uloží do souboru **Uf-s-cislo.txt**, kde: část názvu „**Uf**“ nám říká, že jde o data získaná ze schématu **Uf_.mdl**
 - část „**s**“, že jde o data získaná simulací
 - část „**cislo**“ – je stejná jako u předchozího schématu
- Datový soubor potřebujeme pro program VizData.

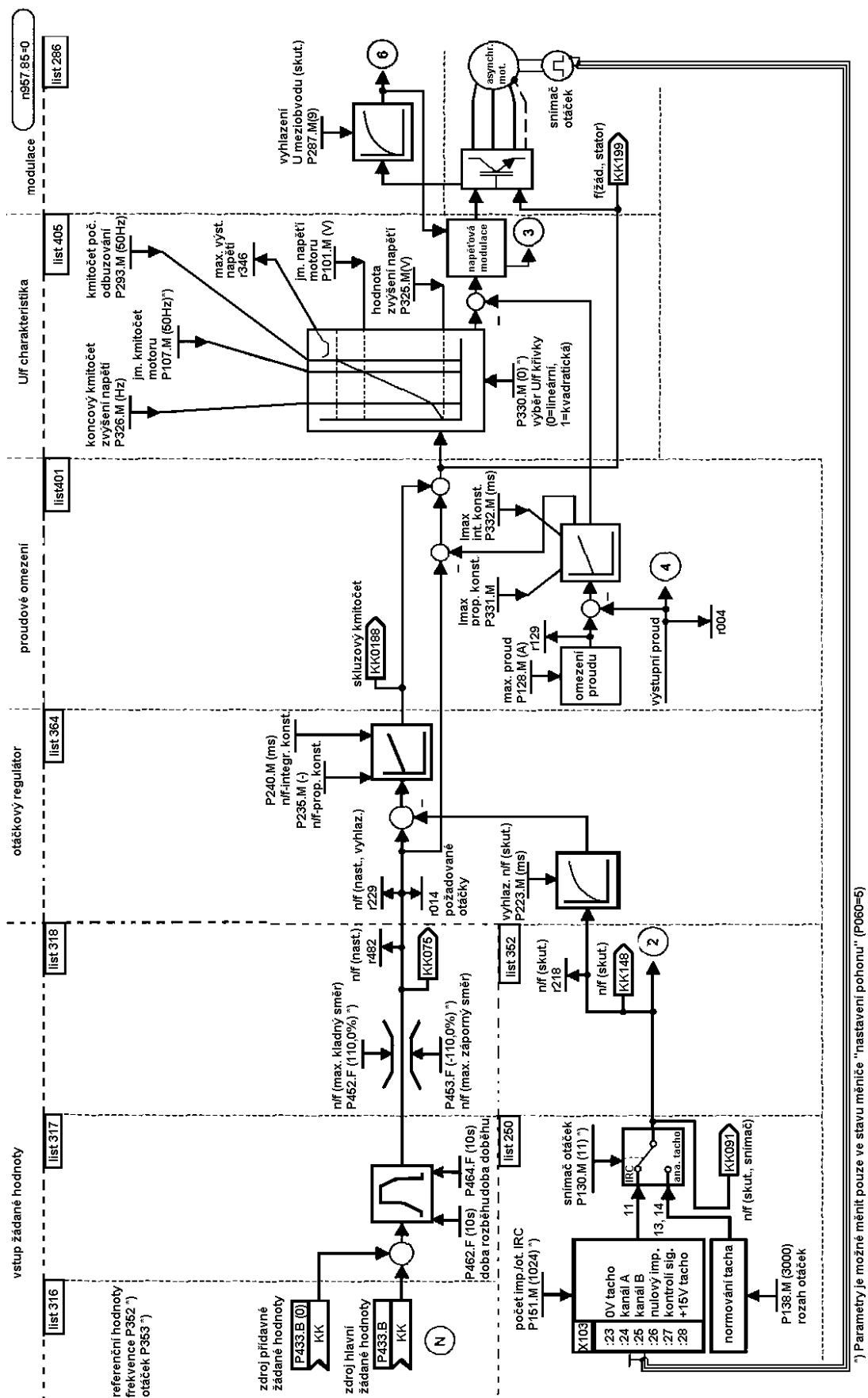


Obr.6.8 - Okno Editor/Debugger – soubor Uloz_data.m

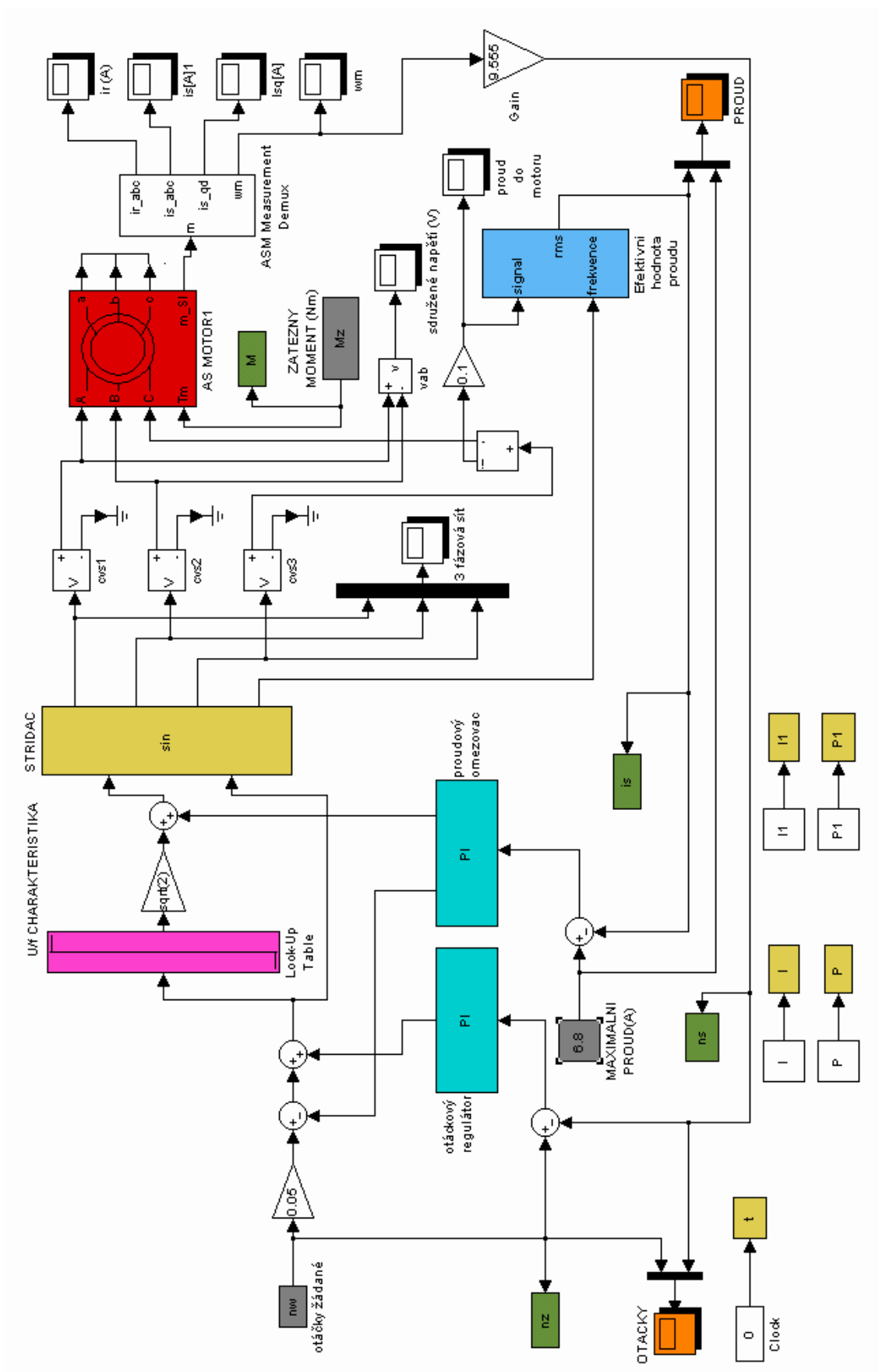
6.1.3 Model U/f řízení s otáčkovým regulátorem měniče SIMOVERT

U/f charakteristika s otáčkovým regulátorem je v případě frekvenčního měniče SIMOVERT nejdokonalejším způsobem skalárního řízení. Pro simulaci U/f řízení se zpětnou otáčkovou vazbou je výchozí parametrizační schéma na obr.6.11. Ze schématu vychází separabilita systému a je jím určena rozlišovací úroveň pro tvorbu modelu (obr.6.12). Ve schématu jsou viditelné zvláště regulační smyčky otáček a proudového omezovače. Některé dílčí informace lze nalézt v Compendiu k měniči SIMOVERT. Rozlišovací úroveň je omezena přístupem k podrobnějším schématům.

Kauzalitu systému U/f řízení se zpětnou otáčkovou vazbou lze popsat takto: Ze vstupující žádané hodnoty otáček je určena odpovídající frekvence. Hlavní větví tato hodnota pokračuje do místa součtového bloku, ve kterém je porovnávána se skutečnou hodnotou frekvence vypočtené ze skutečných rotorových otáček. Hodnota skutečných rotorových otáček je určena inkrementálním snímačem otáček připojeným na hřídel AS motoru. Je-li zjištěna regulační odchylka, zasáhne do řízení otáčkový PI regulátor, který frekvenci sníží nebo případně zvýší. Frekvence upravená otáčkovým regulátorem pokračuje dále k dalšímu součtovému bloku, ve kterém zasahuje v případě překročení maximální hodnoty proudového omezení proudový PI regulátor. Při překročení proudového omezení PI regulátor frekvenci sníží. Hodnota okamžitého proudu je určena z výstupního trojfázového napájecího napětí AS motoru proudovým čidlem měniče. Frekvence korigovaná regulátory pokračuje k U/f charakteristice, která provede přiřazení napětí příslušejícího dané frekvenci. Napětí vystupující z U/f charakteristiky je v případě překročení proudového omezení na výstupu U/f charakteristiky sníženo proudovým PI regulátorem. Tím se dosáhne rychlejší reakce na překročení proudového omezení. Hodnotu výstupního napětí z U/f charakteristiky připraví napěťový meziobvod. Napětí meziobvodu je stejnosměrné a vstupuje do střídače, který je upravuje PWM modulací. PWM modulací se vytvoří, dle frekvence na vstupu do U/f charakteristiky, třífázová napájecí síť AS motoru, s parametry amplitudy a frekvence napětí, určenými na vstupu do bloku střídače.



Obr.6.11 - U/f charakteristika s otáčkovým regulátorem (P100=0)



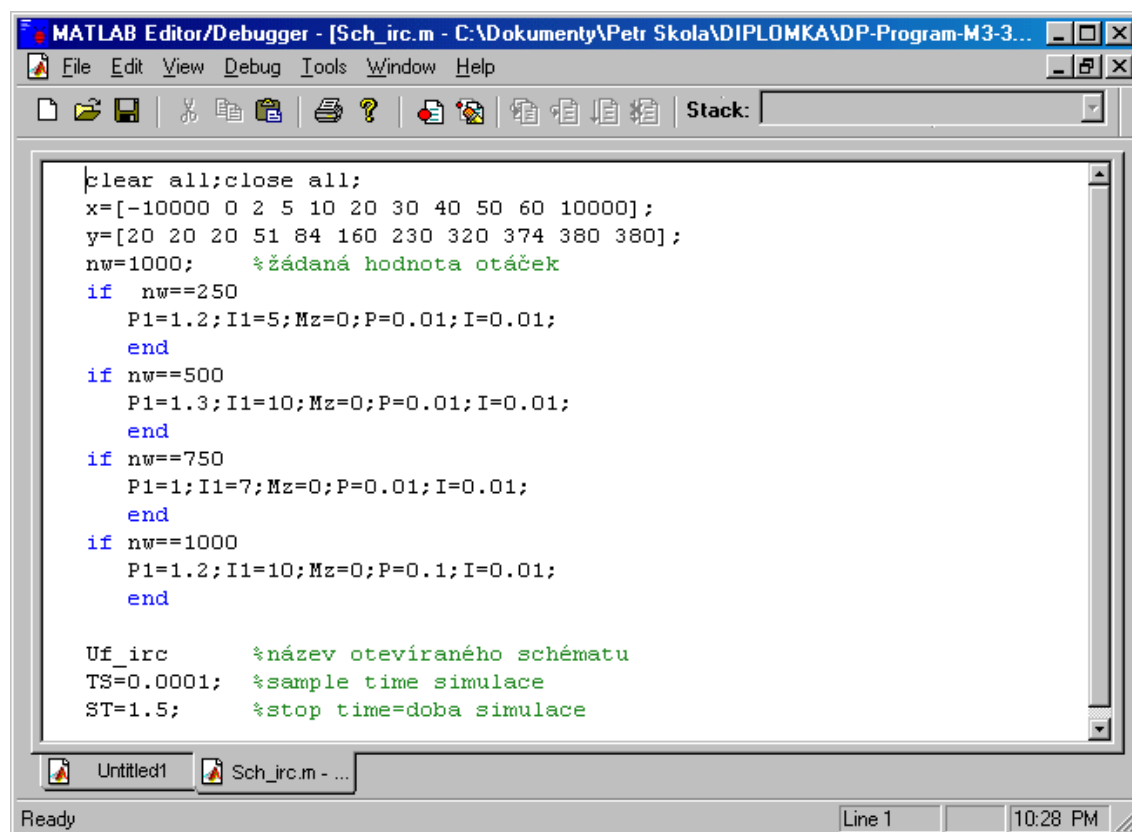
Obr.6.12 – Schéma U/f řízení s otáčkovým regulátorem

Schéma na obr.6.12 sestavené v prostředí Simulink je vytvořeno z obdobných prvků jako v případě předcházejících U/f charakteristik.

Toto schéma najdeme pod souborem simulinku *Uf_irc.mdl*. Dále budeme potřebovat soubory *Sch_irc.m* a *Uloz_data_irc.m*. Tyto 3 soubory musí být uloženy v jednom adresáři. Soubor *Sch_irc.m* obsahuje sadu potřebných parametrů pro běh simulace stejně jako u dvou předchozích U/f charakteristik.

Při spuštění a následné simulaci postupujeme ve stejných krocích jako v předchozích kapitolách s tím rozdílem že:

- v kroku 3. otevřeme soubor *Sch_irc.m*.
- dále pak v otevřeném souboru můžeme kromě již zmíněných parametrů zadat také parametry otáčkového regulátoru: **P** - proporcionální složka a **I** - integrační složka (obr.6.13)



```
clear all;close all;
x=[-10000 0 2 5 10 20 30 40 50 60 10000];
y=[20 20 20 51 84 160 230 320 374 380 380];
nw=1000;    %žádaná hodnota otáček
if nw==250
    P1=1.2;I1=5;Mz=0;P=0.01;I=0.01;
end
if nw==500
    P1=1.3;I1=10;Mz=0;P=0.01;I=0.01;
end
if nw==750
    P1=1;I1=7;Mz=0;P=0.01;I=0.01;
end
if nw==1000
    P1=1.2;I1=10;Mz=0;P=0.1;I=0.01;
end

Uf_irc    %název otevíraného schématu
TS=0.0001; %sample time simulace
ST=1.5;   %stop time=doba simulace
```

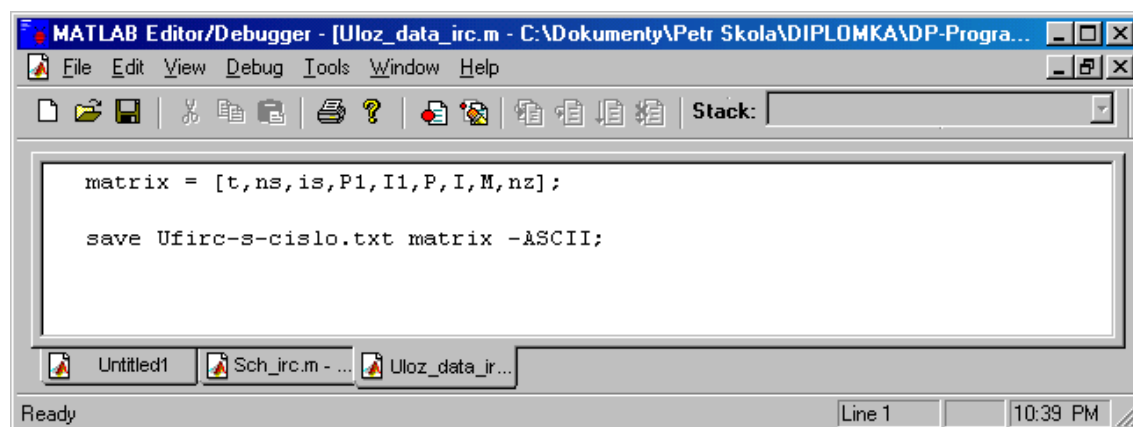
Obr.6.13 - Okno Editor/Debugger – soubor Sch_irc.m

- v kroku 4.se nám po spuštění souboru otevře simulační schéma *Uf_irc.mdl* (viz obr.6.12)

- po provedení simulace následuje krok 6., uložení dat do souboru, zde si ovšem musíme otevřít soubor *Uloz_data_irc.m* (obr.6.14), po spuštění se nám data uloží do souboru *Ufirc-s-cislo.txt*, kde:

- část názvu „*Ufirc*“ nám říká, že jde o data získaná ze schématu *Uf_irc.mdl*
- část „*s*“, že jde o data získaná simulací
- část „*cislo*“ – je stejná jako u předchozího schématu

Datový soubor je nutný pro vizualizační program VizData.



Obr.6.15 - Okno Editor/Debugger – soubor Uloz_data_irc.m

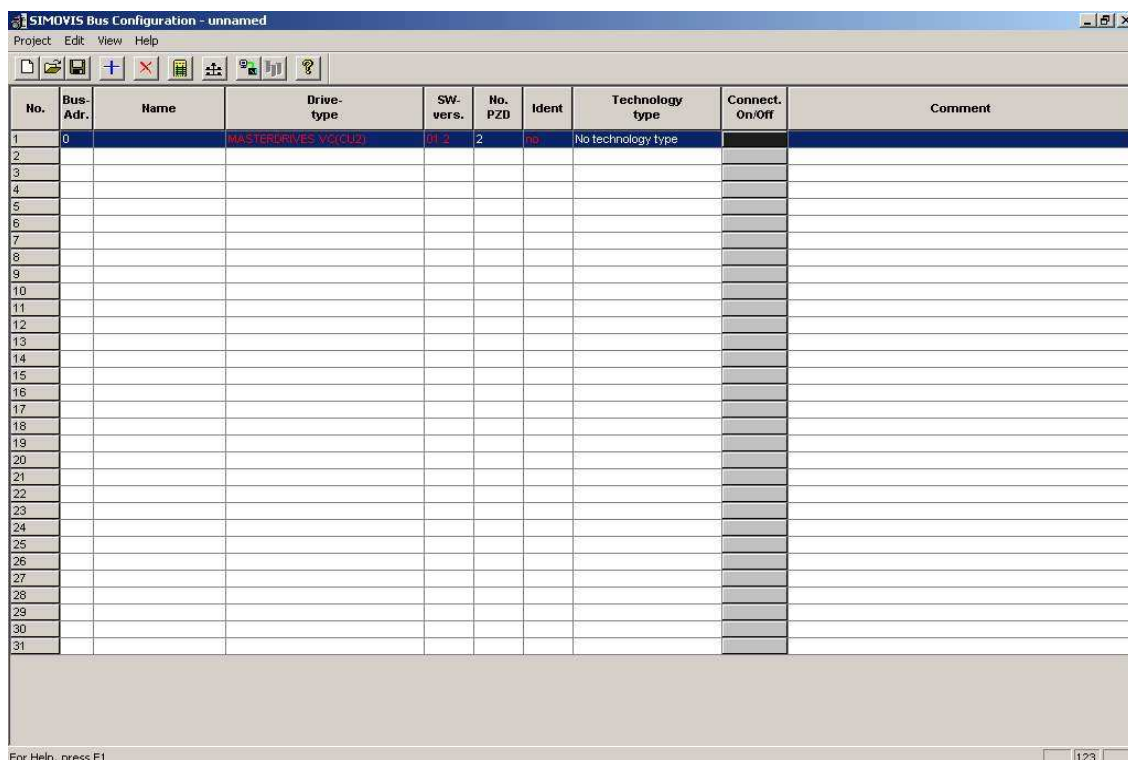
6.2 Měřicí část

Tato část laboratorního cvičení je zaměřena na bližší seznámení s programem SIMOVIS, na princip zadávání parametrů a samotnou práci s tímto programem.

Budeme předpokládat, že dynamický systém v laboratoři TUL je zapojen tak, jak má být: AS motor je zapojen do sítě, frekvenční měnič je funkční a zapnut, PC pracoviště je propojeno s frekvenčním měničem a ten s AS motorem, na PC je nainstalován program SIMOVIS.

SIMOVIS je programové vybavení od firmy SIEMENS. Frekvenční měnič vyžaduje pro svou činnost nastavení parametrů, které ovlivňují jak jeho regulační strukturu, tak i nastavení konstant regulátorů, parametrů ovlivňujících rozběh, definování rozběhových ramp, nastavení proudových omezovačů atd.. Podle typu regulačního schématu se jedná přibližně až o 900 parametrů, které je možné nastavit před samotnou činností měniče, nebo je lze měnit za běhu. K tomuto nastavování lze využít nejen samostatných ovládacích prvků integrovaných přímo na skříní měniče, ale i technologické klávesnice umístěné např. na stole operátora, který pomocí klávesnice mění parametry měniče ze svého pracoviště. Vzhledem k tomu, že měnič také obsahuje

výstupní komunikační porty, lze k této činnosti přímo použít programový prostředek SIMOVIS. Program SIMOVIS vyžaduje ke svému běhu PC s nainstalovaným operačním systémem minimálně Windows 95, s pamětí RAM o velikosti minimálně 16 MB a alespoň 10 MB volného místa na harddisku. Ke komunikaci využívá standardních sériových portů a komunikuje pomocí protokolu USS s měničem, případně při použití vzájemně se softwarem SIMATIC využívá ke komunikaci Profibus. Po spuštění SIMOVISu v prostředí Windows se otevře základní okno SIMOVISU (obr.6.16).



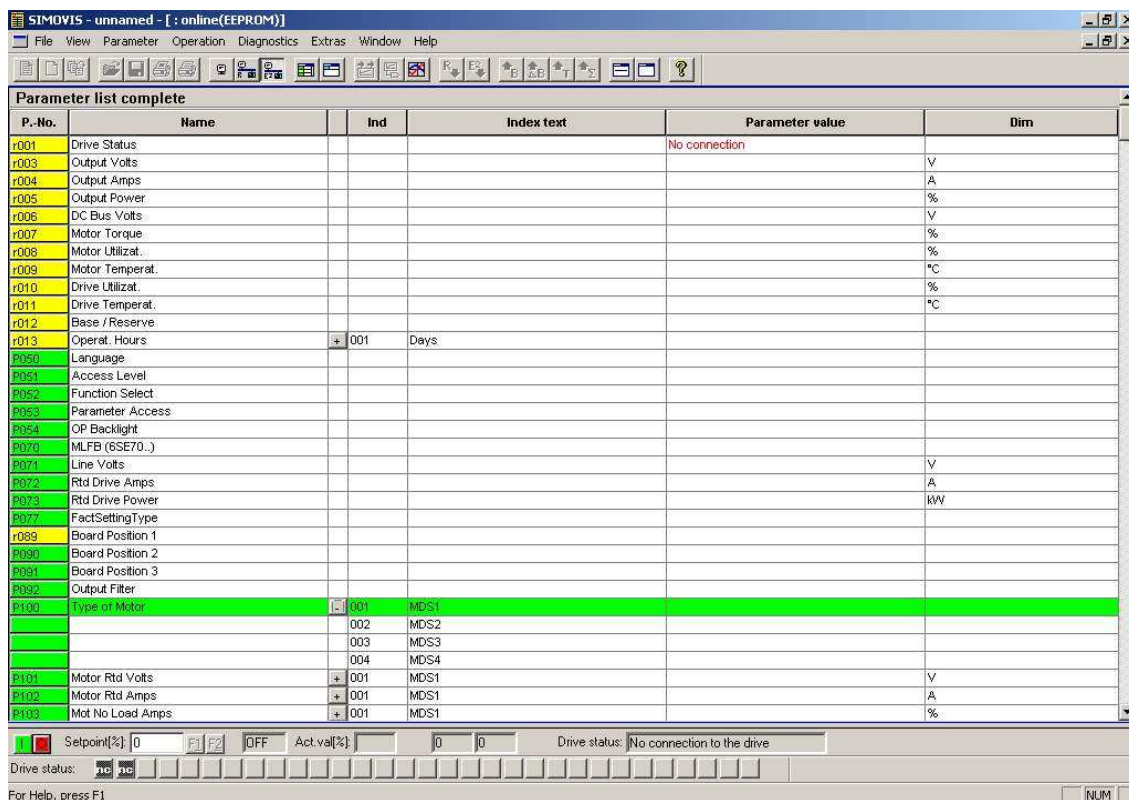
Obr.6.16 - Základní okno SIMOVISU

V tomto okně jsou ikony pomocí nichž se dá upřesnit typ parametrizovaných dat, je to především ikona, sloužící k výběru typu řízení:



Add Drive.

Pro měnič užívaný v laboratořích TUL je nejvhodnějším typem parametrů řízení MASTERDRIVES VC (CU2). Lze vybrat i jiný typ řízení, ale tento výběr přímo ovlivňuje model měniče, používaného společně s programem SIMOVIS. Po provedení výběru se otevře okno parametrů, ve kterém lze upravovat a měnit nastavení parametrů (obr.6.17).



Obr.6.17 - Okno parametrů SIMOVISU

Čísla parametrů se skládají z písmene a třímístného čísla.

Velká písmena P, U, H a L označují parametry u nichž lze změnit hodnotu.

P pro parametry s číslem <1000

H od ≥ 1000 do <2000

U od ≥ 2000 do <3000

L od ≥ 3000

Parametry označené malými písmeny r, n, d, c označují parametry, jejichž hodnotu lze jen číst. Tyto parametry se označují jako „monitorovací“ parametry.

r pro parametry s číslem <1000

d od ≥ 1000 do <2000

n od ≥ 2000 do <3000

c od ≥ 3000

Po provedení změn lze vlastní běh aktivovat ikonou vypnuto-zapnuto, která se nachází v levém dolním rohu okna SIMOVISu. Počítač poté zahájí komunikaci s měničem, který ověří nastavení komunikace, případně ji upraví. Programový prostředek SIMOVIS neslouží však jen k pouhému parametrizování, případně ke změně

struktury měniče, je to nástroj použitelný také k vizualizaci činnosti měniče a také k diagnostice činnosti měniče.

SIMOVIS nabízí funkce tisku parametrů, nebo pouze změřených parametrů, nastavení parametrů motoru, přednastavení sady parametrů, např. pro U/f řízení v textilním průmyslu. Posledně jmenovaná funkce velmi zjednodušuje a zpřehledňuje úpravy parametrů, protože se uživatel nemusí zabývat velkým množstvím parametrů, které pro určitý typ řízení není potřeba zobrazovat. V SIMOVISu lze také nastavovat parametry komunikace mezi měničem a PC (typ rozhraní, protokolu, atd.), a také ukládat vypracované projekty nastavení měniče.

Nastavení a ovládání měniče SIMOVERT pomocí počítače PC a programu SIMOVIS

Uvedeme měnič do stavu „nastavení pohonu“ - **P060 = 5**

Vybereme požadovaný mód řízení a regulace - **P100 = x**

P100 – Mód řízení a regulace (list r0 až r5)

Tímto parametrem vybíráme z přednastavených módů řízení a regulace.

Vybereme požadovaný mód řízení a regulace - **P100 =**

0: řízení dle charakteristiky U/f, regulace otáček s impulsním čidlem otáček

(Skalární řízení otáček regulované inkrementálním čidlem otáček, viz obr.6.11)

1: řízení dle charakteristik U/f

(Skalární řízení pohonu s regulací skluzu, viz obr.6.9)

2: řízení dle charakteristiky U/f (aplikace v textilním průmyslu)


(Skalární řízení bez regulace skluzu, viz obr.6.1)

P462 - Doba rozběhu

Nastavíme na 0 s, chceme vidět okamžitou reakci.

P357 - Vzorkovací čas

Doba měření je závislá na vzorkovacím času (implicitně nastaven 1,2 ms – my tuto hodnotu ponecháme kvůli vizualizačnímu programu, jinak je rozsah nastavitelných hodnot: 0,8 - 4,0 ms) a zvoleném měřítku. Měřítka y osy je pro každý zobrazovaný signál jiné - zobrazeno je měřítko jen pro označený signál.

Vybereme signály jejichž závislosti budeme chtít zobrazit do grafu. Každý signál uvedený ve schématu je označen číselným kódem s prefixem Kxxx (16bit. slovo), případně KKxxx (32bit. slovo). Všechny tyto hodnoty jsou vztaženy k referenčním hodnotám (avšak ne vždy se musí jednat o jmenovité hodnoty pohonu) jednotlivých signálů a uváděny v %. Menu otevřeme ikonou SETTINGS .

K0184 – Isq(act) – momentotvorná složka proudu

K0241 – Torque(act) – moment motoru

K0024 – Motor Torque – také moment motoru (jen u vektorového řízení)

KK0199 – F(set,stator) – žádaná hodnota frekvence (resp. Otáček – 100% odpovídá frekvenci 50 Hz a otáčkám 1000 ot/min)


KK0091 – Meas'dRot.Speed - signál z irc čidla bez vyhlazení (je zde možnost nahrazení přidáním volného bloku pro vyhlazení na výstup otáček – průběhy jsou přijatelnější)

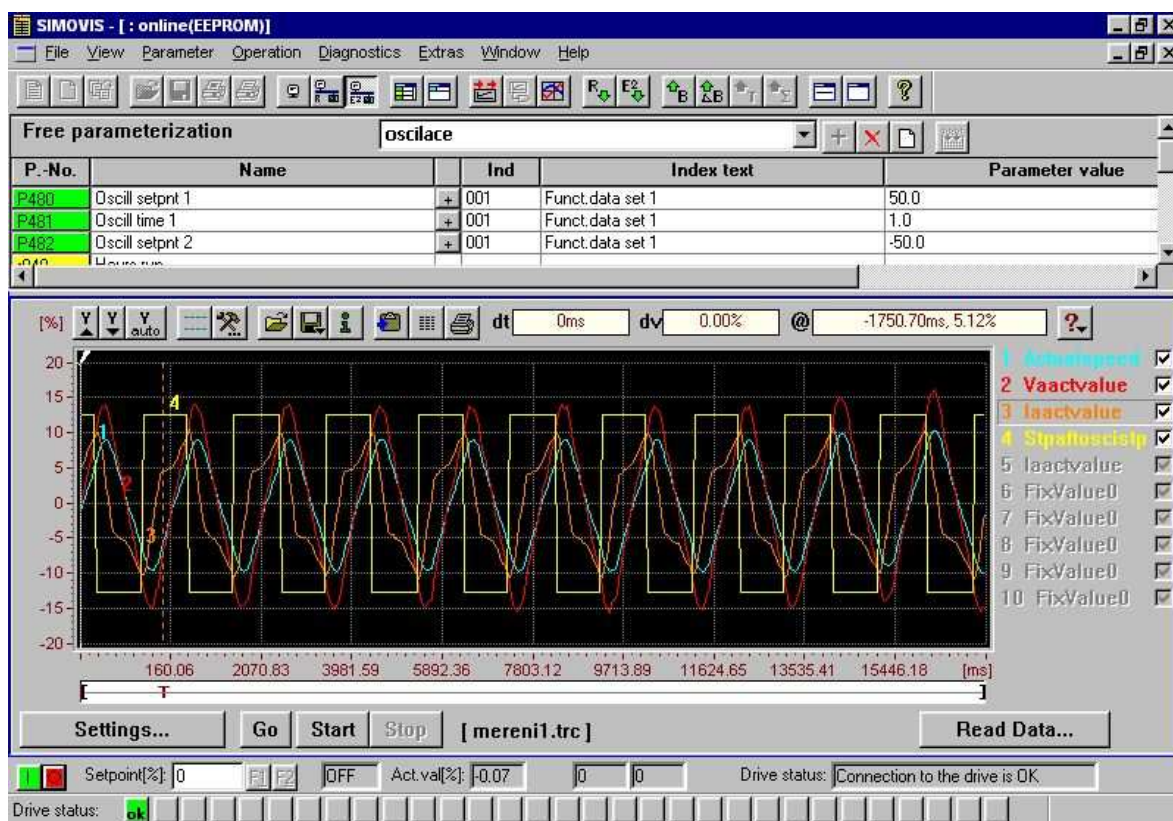
K0021 – Output Volts – výstupní napětí střídače

K0238 – Phase 1 Amps – výstupní proud fáze U

POZOR!!! Je nesmírně důležité zachovat posloupnost zobrazovaných dat výše uvedených, poněvadž se nám v tomto pořadí budou i ukládat do datového souboru viz. pokračování a pak se s takto uloženým souborem pracuje v programu VizData.

Analogové výstupy a zobrazení aktuálních hodnot je na schématu v příloze č.1.

Program SIMOVIS umožňuje uložení zaznamenaných dat do vlastního formátu *.trc. Tento formát uloží naměřené charakteristiky, nastavení vybraných signálových konektorů a měřítek os. Dále export naměřených charakteristik do grafického formátu WMF (Windows Media File) a také do, pro nás důležitého, textového souboru *.txt, ten my potřebujeme pro program VizData. Možnosti uložení dat se objeví po kliknutí na ikonu SAVE .



Obr.6.18 - Ukázka prostředí programu SIMOVIS po dokončení měření

K již zmíněnému ukládání dat do souboru si musíme krátce něco říci, je nezbytně nutné aby ukládané soubory měly tvar:

pro **P100=0** - *Ufirc-m-cislo.txt* kde část názvu „*Ufirc*“ nám o souboru říká, že data jsou získána z měření na módu *Skalární řízení otáček regulované inkrementálním čidlem otáček*, část „*-m-*“ naznačuje, že data jsou získána měřením na reálném pohonu a část „*cislo*“ nám udává číslo měření (1, 2, ..., 100).

pro **P100=1** - *Uf-m-cislo.txt* kde část názvu „*Uf*“ nám o souboru říká, že data jsou získána z měření na módu *Skalární řízení pohonu s regulací skluzu*, část „*-m-*“ je stejná jako u předchozího souboru a stejně tak část „*cislo*“.

pro **P100=2** - *Uftex-m-cislo.txt* kde část názvu „*Uftex*“ nám o souboru říká, že data jsou získána z měření na módu *Skalární řízení bez regulace skluzu*, část „*-m-*“ je stejná jako u předchozích souborů tak jako část „*cislo*“.

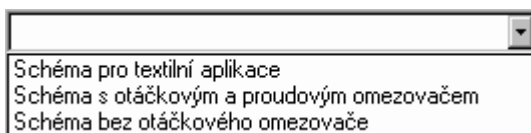
Část „*cislo*“ v názvu souboru je velice důležitá z hlediska vizualizace dat v programu VizData, jak bylo předesláno na konci kap.6.1.1.


6.3 Vyhodnocovací část



Nyní bychom již měli mít sadu souborů ze simulační části cvičení a z části věnované měření. Takže nám nic nebrání v porovnání získaných dat. K tomuto účelu nám poslouží, v předchozích kapitolách již několikrát zmiňovaný, vizualizační program **VizData.exe** (součást přiloženého CD). Tento program byl vytvořen v prostředí Borland Delphi (zdrojový kód je v příloze č.2) a jde o jednoduchý grafický interface, který funguje nezávisle na programech SIMOVIS a MATLAB. Toto je hlavní důvod vytvoření programu VizData – funkčnost na jakémkoliv PC (s disketovou jednotkou kvůli načtení dat a nahrání samotného programu) bez nutnosti instalovat SIMOVIS nebo MATLAB.

Hlavní podmínkou správného užití programu je nezbytná přítomnost námi vytvořených datových souborů a souboru programu v jednom adresáři. Pokud je toto splněno pak můžeme program spustit.

Zobrazí se nám okno programu (obr.6.19), vidíme zde několik tlačítek, 2 oblasti grafů a několik rolovacích menu. Abychom si mohli porovnat naše data, je třeba se s ovládáním programu nejprve seznámit. Nejprve si musíme vybrat schéma (druh skalárního řízení) a to v rolovacím menu, na výběr tu máme:

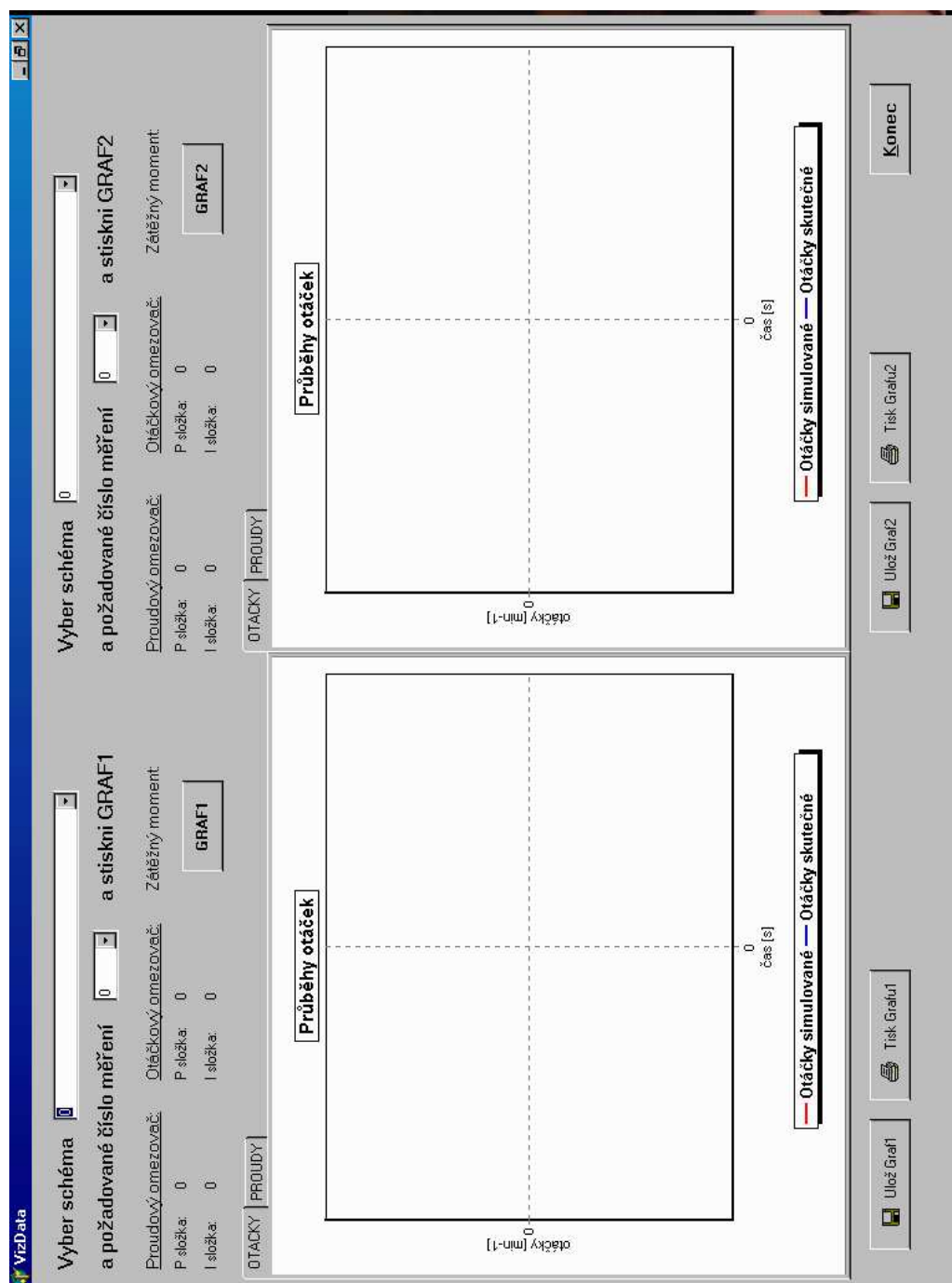


zvolíme si např. Schéma pro textilní aplikace , poté je třeba zvolit v dalším rolovacím menu požadované číslo měření, zvolíme třeba 1.

Pak je třeba stisknout tlačítko GRAF1 , a po jeho stisknutí by, pokud je vše v pořádku, levá polovina obrazovky měla vybadat přibližně tak, jak je vidět na obrázku 6.20. Zobrazily se nám průběhy otáček a proudů, na průběhy proudů se podíváme jednoduše kliknutím myši na záložku s názvem proudy .

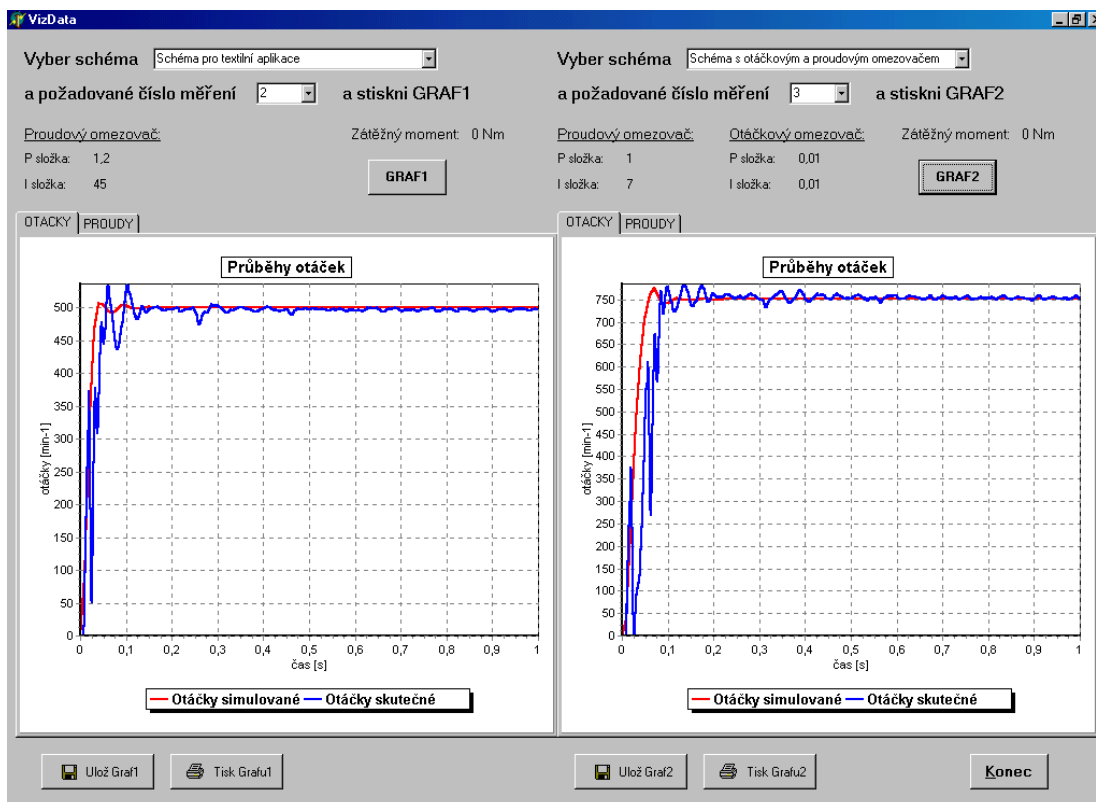
Nad průběhy jsou vypsány hodnoty proudového omezovače, zátěžný moment motoru a u schématu s otáčkovým regulátorem také hodnoty otáčkového regulátoru. Stejně bychom postupovali v případě pravé strany okna s tím rozdílem, že je třeba stisknout tlačítko GRAF2. Dále jsou v okně programu tlačítka Ulož Graf1(resp.2) a Tisk Grafu1(resp.2). Tlačítko Ulož Graf1(2) nám uloží aktuální průběh, buď Otáčky nebo Proudů, do našeho adresáře (ze kterého jsme program spouštěli) jako obrázek ve formátu **gr1_ot-tex-2.bmp** nebo **gr1_pr-tex-2.bmp** pro Grafy1 resp. **gr2_ot-tex-2.bmp** nebo **gr2_pr-tex-2.bmp** pro

Grafy2, opět vysvětlíme co nám říká název souboru: „**gr1(gr2)**“ – jde o obrázek z grafu1(2), „**ot(pr)**“ – průběh otáček(proudů), „**tex(irc, nic)**“ – jde o schéma pro textilní aplikace (s otáčkovým regulátorem nebo jen U/f). Tlačítko Tisk Grafu1(2) nám vytiskne aktuální průběh, buď Otáčky nebo Proudů, na dostupné nainstalované tiskárně.

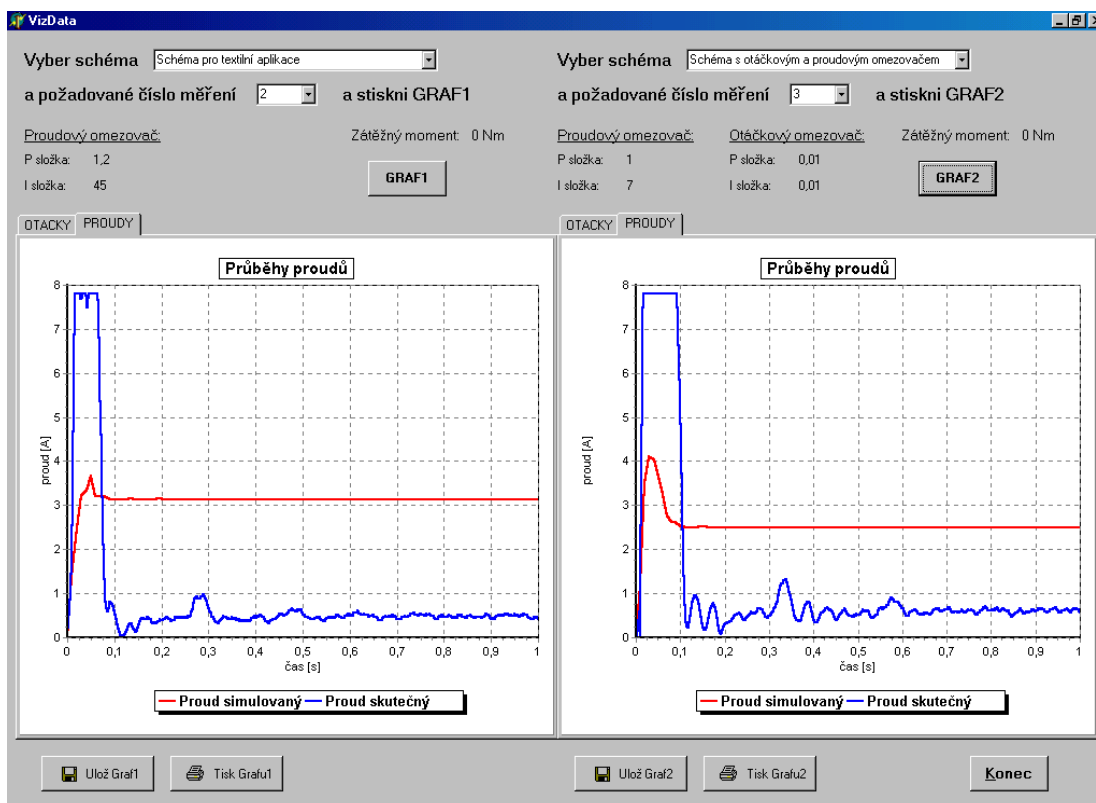


Obr.6.19 – Okno programu VizData po spuštění

Jestliže jsme postupovali podle předešlého popisu správně, tak by výsledné okno mohlo vypadat pro otáčky (viz obr.6.20) a pro proudy (viz obr.6.21) obdobně.



Obr.6.20 – Ukázka okna programu VizData pro otáčky

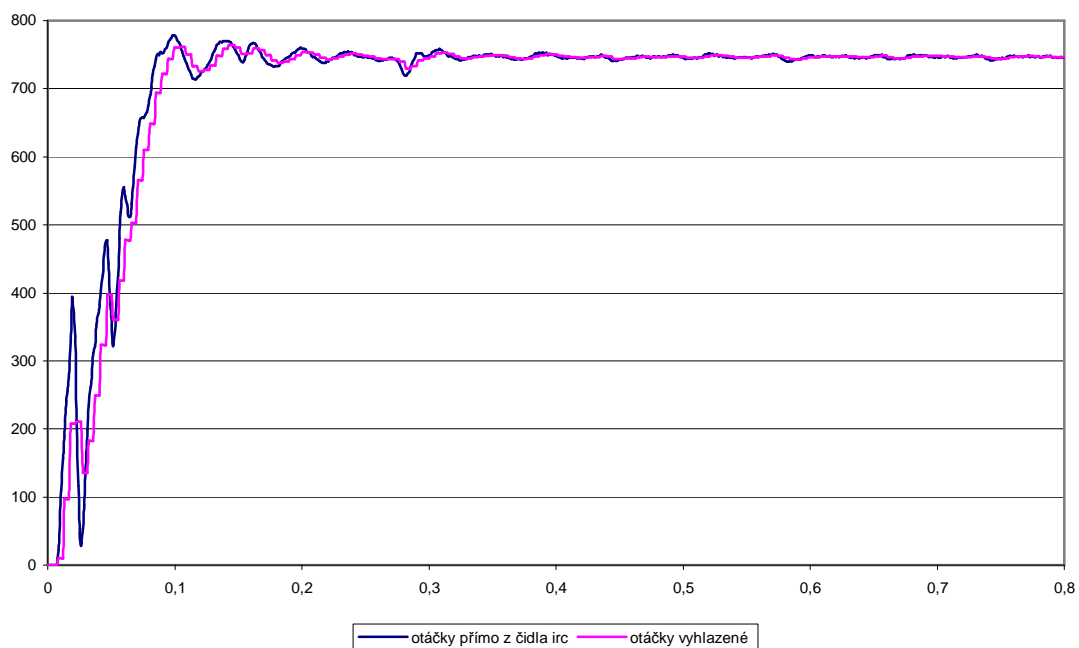


Obr.6.21 – Ukázka okna programu VizData pro proudy

Závěrem cvičení je zapotřebí vyhodnotit průběhy a konstatovat do jaké míry jsou si odpovídající průběhy podobné případně shodné a do jaké se od sebe odlišují.

7 ZHODNOCENÍ

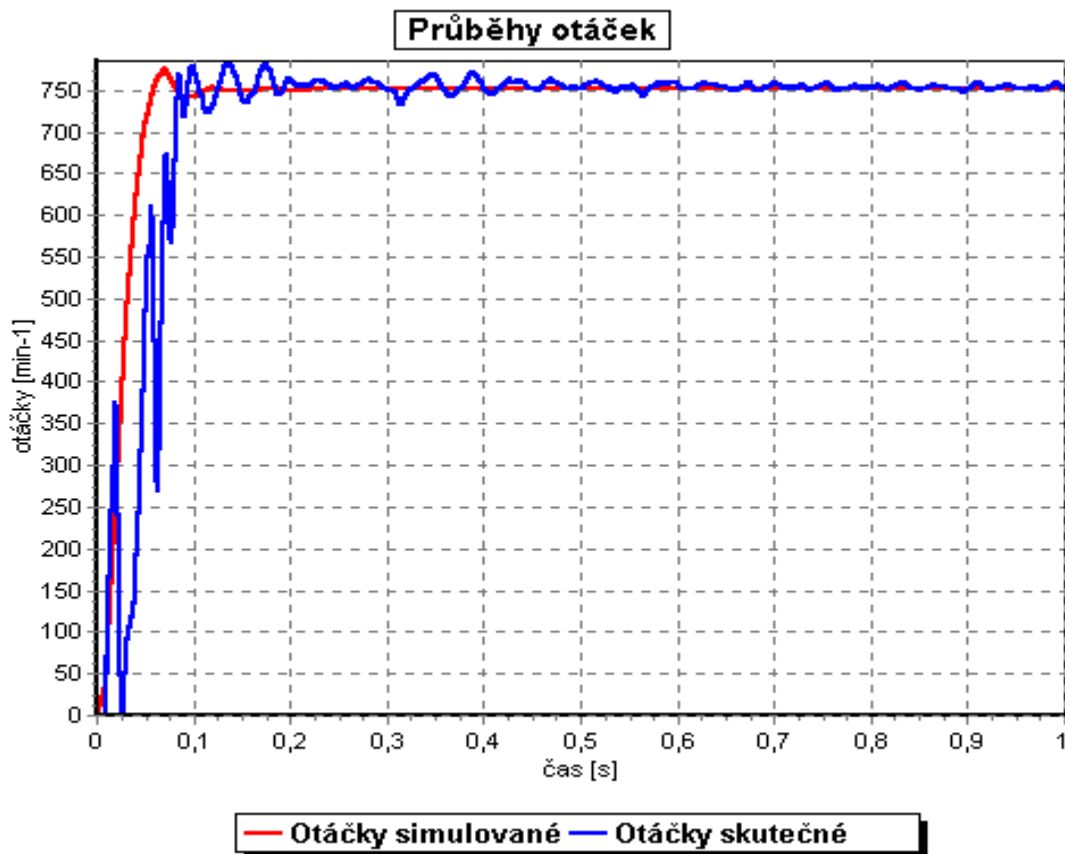
Na následujících 3 obrázcích je vidět do jaké míry se shodují průběhy otáček na vybraném schématu skalárního řízení s proudovým a otáčkovým regulátorem při žádaných otáčkách 750 ot/min.



Obr.7.1 – Průběh otáček získaný měřením na reálném pohonu



Obr.7.2 – Průběh otáček získaný simulací



Obr.7.3 – Průběhy otáček získaných simulací a měřením v programu VizData

Je patrné, že si jsou průběhy otáček podobné. Avšak při vytváření simulací se vyskytly určité nedostatky. Na vstupu do bloku AS motoru vycházely nereálné proudy. Problém byl dočasně vyřešen užitím korekční konstanty, která proud snižuje na reálnou hodnotu. Hodnota konstanty byla určena z naměřené hodnoty proudu při jmenovitých otáčkách skutečného AS motoru v chodu naprázdno. Tato hodnota se však jeví jako nevyhovující pro celý rozsah otáček. Dále je možné, že blok AS motoru není ideální pro další pokračování simulací a stálo by za úvahu navrhnout vlastní blok AS motoru, ve kterém by byly čitelné algoritmy a průběhy veškerých důležitých veličin. Existuje také možnost, že vytvořená simulační schémata řízení nejsou natolik shodná se silovými schémata U/f řízení firmy SIEMENS. Pro další směr simulací bych doporučil prověřit, zda spolu korespondují silová schémata měniče a vytvořená simulační schémata. Program VizData je založen na schopnosti obou již zmiňovaných programů (SIMOVIS a MATLAB) ukládat data do textových souborů a jejich následného zobrazování v grafech. Možné zlepšení bych viděl ve vytvoření programu komunikujícího s oběma programy v reálném čase.

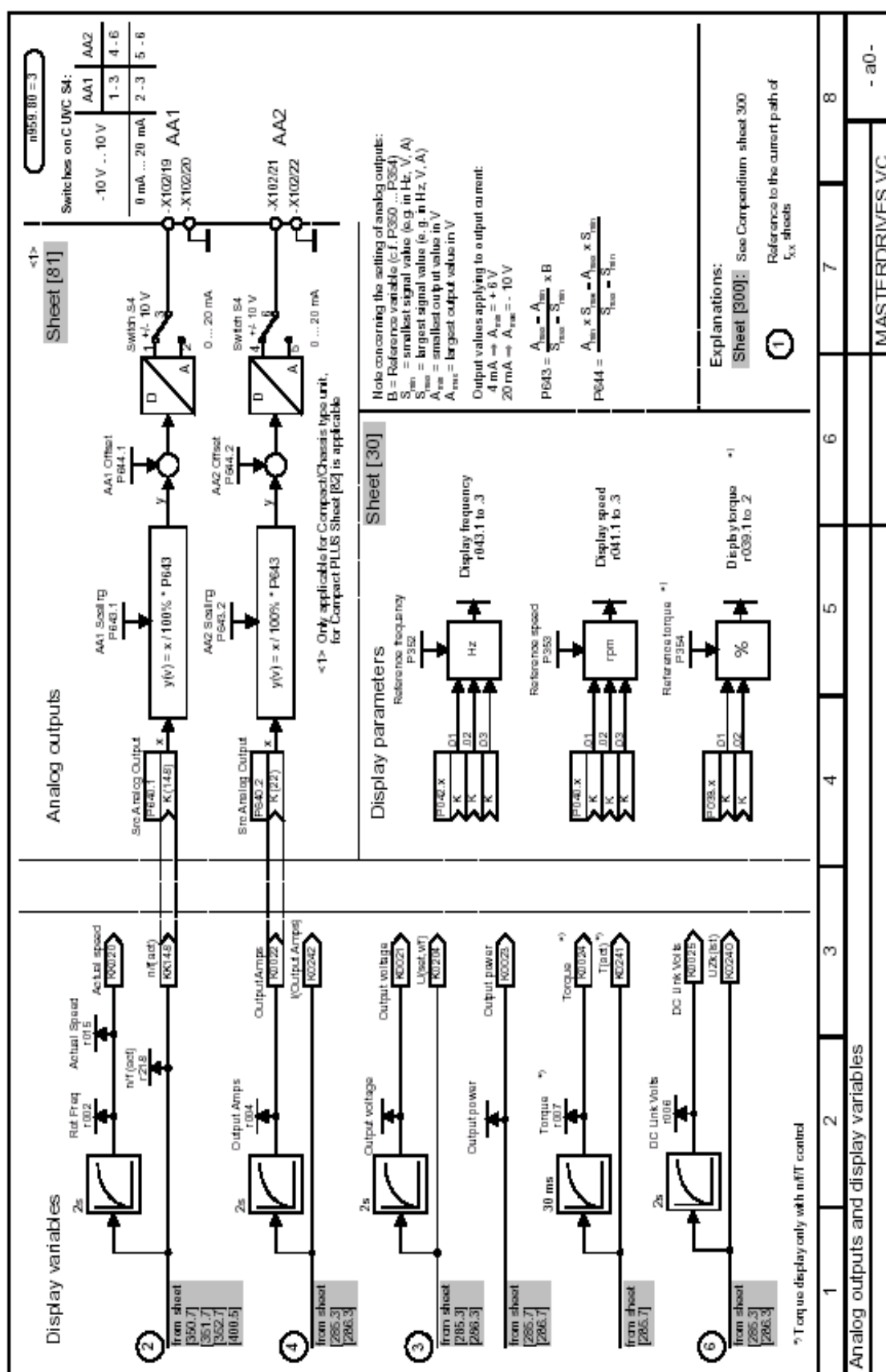
8 ZÁVĚR

Při vytváření této studie bylo vycházeno ze silových schémat frekvenčního měniče a ze simulačních schémat navržených v projektu Řízení elektrického pohonu s měničem SIMOVERT nadřazeným počítačem, který byl zpracován jako Diplomová práce Ing. Tesařem. Snahou bylo sestavit modely, které by výchozím schématům co nejvíce odpovídaly. Cílem bylo vytvoření vizualizačního programu i s návodem na jeho obsluhu, bez vysokých požadavků na Hardware a Software PC, na kterém bude používán a pokus o začlenění programu do výuky elektrických pohonů návrhem laboratorního cvičení. Struktura programu VizData pramení ze znalostí autora ve spojení s použitou literaturou (kompletní zdrojový kód programu ve v příloze 2 a na přiloženém CD). Součástí je také přiblížení práce se simulačním programem Matlab/Simulink a vytvořenými simulačními schématy. Dotkli jsme se zde také problematiky práce s programem SIMOVIS.

9 SEZNAM POUŽITÝCH ZDROJŮ

- [1] Tesař M.: *Řízení elektrického pohonu s měničem SIMOVERT nadřazeným počítačem*, Diplomová práce TUL, KEL 2002
- [2] Pavelka J., Čerovský Z., Javůrek J.: *Elektrické pohony*, ČVUT Praha 1999
- [3] Bulgakov A.A.: *Řízení asynchronních motorů měniči frekvence*, SNTL Praha 1989
- [4] Kadlec V.: *Učíme se programovat v Delphi a jazyce object pascal*, Computer Press Praha 2001
- [5] Sedláček J., Slaba J.: *Delphi v kostce*, BEN Praha 2000
- [6] Noskovič P.: *Modelování a identifikace systémů*, MONTANEX a.s. Ostrava 1999
- [7] Křivý I., Kindler E.: *Simulace a modelování*, Ostravská univerzita Ostrava 2001
- [8] Pokorný P.: *Řízení elektrického pohonu SIEMENS změnou programového vybavení měničů SIMOVERT a SIMOREG*, Ročníkový projekt TUL, KEL 2003
- [9] Kalaš D., Jucha I.: *Matlab/Simulink*, STU Bratislava 1996
- [10] Uhlíř I. a kol.: *Elektrotechnika*, ČVUT Praha 1998
- [11] *SIMOVERT MASTERDRIVES Vector Control návod k obsluze a údržbě*, BA 9807-000-348 BB, SIEMENS AG 1998
- [12] www.siemens.de - internetová adresa firmy SIEMENS
- [13] www.humusoft.cz - internetová adresa firmy HUMUSOFT s.r.o.
- [14] www.mathworks.com - internetová stránka věnovaná MATLAB/SIMULINK

PŘÍLOHY



Příloha č.2

```
unit PetrBubenDP;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, TeEngine, Series, ExtCtrls, TeeProcs, Chart, ComCtrls, Buttons;

type
  TForm1 = class(TForm)
    Label1: TLabel;
    ComboBox2: TComboBox;
    Button1: TButton;
    Label2: TLabel;
    Button2: TButton;
    PageControl1: TPageControl;
    TabSheet1: TTabSheet;
    TabSheet2: TTabSheet;
    Chart1: TChart;
    Series1: TLineSeries;
    Series2: TLineSeries;
    Chart2: TChart;
    Series3: TLineSeries;
    Series4: TLineSeries;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    PageControl2: TPageControl;
    TabSheet3: TTabSheet;
    TabSheet4: TTabSheet;
    Chart3: TChart;
    Series5: TLineSeries;
    Series6: TLineSeries;
    Chart4: TChart;
    Series7: TLineSeries;
    Series8: TLineSeries;
    ComboBox1: TComboBox;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    Label9: TLabel;
    Label10: TLabel;
    Label11: TLabel;
    Label12: TLabel;
    Label13: TLabel;
    Label14: TLabel;
    Label15: TLabel;
    Label16: TLabel;
    Label17: TLabel;
    Label18: TLabel;
    Label19: TLabel;
    Label20: TLabel;
    Label21: TLabel;
    Label22: TLabel;
    Label23: TLabel;
    Label24: TLabel;
    Label25: TLabel;
    Label26: TLabel;
    ComboBox3: TComboBox;
    ComboBox4: TComboBox;
    Button3: TButton;
    BitBtn3: TBitBtn;
    BitBtn4: TBitBtn;
    Label27: TLabel;
    Label28: TLabel;
    Label29: TLabel;
    Label30: TLabel;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
  end;

end.
```

```

    procedure BitBtn1Click(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure BitBtn3Click(Sender: TObject);
    procedure BitBtn4Click(Sender: TObject);

private
    { Private declarations }
public
    { Public declarations }
end;

var
    Form1: TForm1;

implementation

{$R *.DFM}

procedure TForm1.Button1Click(Sender: TObject);
begin
    Application.Terminate;
end;

procedure TForm1.Button2Click(Sender: TObject);
type
    TSoubor = TextFile ;
    Matice = array [0..20,0..3000] of real;

var
    Soubor: TSoubor;
    C: char;
    R: real;
    M: Matice;
    Q,K,I,Y,Z: integer;
    cislo: string;

begin
    if not ((ComboBox2.Text = 'Schéma pro textilní aplikace')or
        (ComboBox2.Text = 'Schéma s otáčkovým a proudovým omezovačem')or
        (ComboBox2.Text = 'Schéma bez otáčkového omezovače')) then
        ShowMessage('Prosím zvolte simulační schéma');

    if ComboBox2.Text = 'Schéma pro textilní aplikace' then
        begin
            begin
                Chart1.SeriesList.series[0].Clear;
                Chart2.SeriesList.series[0].Clear;
                Chart1.BottomAxis.Automatic:=false;
                Chart1.BottomAxis.Minimum:=0;
                Chart1.BottomAxis.Maximum:=1;
                Chart2.BottomAxis.Automatic:=false;
                Chart2.BottomAxis.Minimum:=0;
                Chart2.BottomAxis.Maximum:=1;
                Chart2.LeftAxis.AutomaticMaximum:=false;
                Chart2.LeftAxis.Maximum:=8;
                AssignFile(Soubor, 'Uftex-s-'+ComboBox1.Text+'.txt');
                Reset(Soubor);
                cislo:='';
                K:=3;
                Y:=2;
                Z:=0;
                while not eof(Soubor) do
                    begin
                        Read(Soubor,C);
                        if C='.' then
                            C:=',';
                        I:=ord(C);
                        if (((I>47) and (I<58))or (I=44)or (I=101) or (I=43) or (I=45)) and
                            ((K=0)or (K=3))) then
                            begin
                                cislo:=cislo+C;
                                K:=0;
                            end;
                    end;
            end;
        end;
    end;

```

```

        if ((I=32) and (K<>3)) then
            begin
                Q:=ord(cislo[1]);
                if Q<>45 then
                    cislo:=chr(43)+cislo;
                R:=strtofloat(cislo);
                M[Y,Z]:=R;
                cislo:='';
                K:=3;
                R:=0;
                Y:=Y+1;
            end;
        if (I=10) then
            begin
                Chart1.SeriesList.Series[0].AddXY(M[2,Z],M[3,Z]);
                Chart2.SeriesList.Series[0].AddXY(M[2,Z],M[4,Z]);
                Y:=1;
                Z:=Z+1;
            end;
        end;
        CloseFile(Soubor);
        Label7.Caption:=FloatToStr(M[5,3]);
        Label8.Caption:=FloatToStr(M[6,3]);
        Label9.Visible:=false;
        Label10.Visible:=false;
        Label11.Visible:=false;
        Label12.Visible:=false;
        Label13.Visible:=false;
        Label28.Caption:=FloatToStr(M[7,3])+ ' Nm';
    end;
begin
    Chart1.SeriesList.series[1].Clear;
    Chart2.SeriesList.series[1].Clear;
    AssignFile(Soubor, 'Uftex-m-'+ComboBox1.Text+'.txt');
    Reset(Soubor);
    cislo:='';
    K:=3;
    Y:=1;
    Z:=1;
    while not eof(Soubor) do
        begin
            Read(Soubor,C);
            I:=ord(C);
            if (((I>47) and (I<58))or (I=46)or (I=43) or (I=45))and ((K=0)or (K=3)))
then
                begin
                    if I=46 then
                        C:=',';
                    cislo:=cislo+C;
                    K:=0;
                end;
            if ((I=32) and (K<>3)) then
                begin
                    Q:=ord(cislo[1]);
                    if Q<>45 then
                        cislo:=chr(43)+cislo;
                    R:=strtofloat(cislo);
                    if Y=7 then
                        begin
                            R:=R*0.0012;
                            M[Y,Z]:=R;
                        end;
                    if Y=11 then
                        begin
                            R:=R*10;
                            M[Y,Z]:=R;
                        end;
                    if Y=8 then
                        begin
                            R:=R*0.039;
                            M[Y,Z]:=R;
                        end;
                    cislo:='';
                    K:=3;
                    R:=0;
                    Y:=Y+1;
                end;
            end;
        end;
    end;
end;

```

```

        if (I=10) then
            begin
                Chart1.SeriesList.Series[1].AddXY(M[7,Z],M[11,Z]);
                Chart2.SeriesList.Series[1].AddXY(M[7,Z],M[8,Z]);
                Y:=6;
                Z:=Z+1;
            end;
        end;
        CloseFile(Soubor);
    end;
end;
if ComboBox2.Text = 'Schéma s otáčkovým a proudovým omezovačem' then
begin
    begin
        Chart1.SeriesList.series[0].Clear;
        Chart2.SeriesList.series[0].Clear;
        Chart1.BottomAxis.Automatic:=false;
        Chart1.BottomAxis.Minimum:=0;
        Chart1.BottomAxis.Maximum:=1;
        Chart2.BottomAxis.Automatic:=false;
        Chart2.BottomAxis.Minimum:=0;
        Chart2.BottomAxis.Maximum:=1;
        Chart2.LeftAxis.AutomaticMaximum:=false;
        Chart2.LeftAxis.Maximum:=8;
        AssignFile(Soubor,'Ufirc-s-'+ComboBox1.Text+'.txt');
        Reset(Soubor);
        cislo:='';
        K:=3;
        Y:=2;
        Z:=0;
        while not eof(Soubor) do
            begin
                Read(Soubor,C);
                if C='.' then
                    C:=',';
                I:=ord(C);
                if (((I>47) and (I<58))or (I=44)or (I=101) or (I=43) or (I=45)) and
((K=0)or (K=3)) then
                    begin
                        cislo:=cislo+C;
                        K:=0;
                    end;
                if ((I=32) and (K<>3)) then
                    begin
                        Q:=ord(cislo[1]);
                        if Q<>45 then
                            cislo:=chr(43)+cislo;
                        R:=strtofloat(cislo);
                        M[Y,Z]:=R;
                        cislo:='';
                        K:=3;
                        R:=0;
                        Y:=Y+1;
                    end;
                if (I=10) then
                    begin
                        Chart1.SeriesList.Series[0].AddXY(M[2,Z],M[3,Z]);
                        Chart2.SeriesList.Series[0].AddXY(M[2,Z],M[4,Z]);
                        Y:=1;
                        Z:=Z+1;
                    end;
                end;
                CloseFile(Soubor);
                Label7.Caption:=FloatToStr(M[5,3]);
                Label8.Caption:=FloatToStr(M[6,3]);
                Label9.Visible:=true;
                Label10.Visible:=true;
                Label11.Visible:=true;
                Label12.Visible:=true;
                Label13.Visible:=true;
                Label12.Caption:=FloatToStr(M[7,3]);
                Label13.Caption:=FloatToStr(M[8,3]);
                Label28.Caption:=FloatToStr(M[9,3])+ ' Nm';
            end;
        begin
            Chart1.SeriesList.series[1].Clear;
            Chart2.SeriesList.series[1].Clear;

```

```

AssignFile(Soubor, 'Ufirc-m-' + ComboBox1.Text + '.txt');
Reset(Soubor);
cislo:='';
K:=3;
Y:=1;
Z:=1;
while not eof(Soubor) do
begin
  Read(Soubor,C);
  I:=ord(C);
  if (((I>47) and (I<58))or (I=46)or (I=43) or (I=45))and ((K=0)or (K=3)))
then
begin
  if I=46 then
  C:',';
  cislo:=cislo+C;
  K:=0;
end;
if ((I=32) and (K<>3)) then
begin
  Q:=ord(cislo[1]);
  if Q<>45 then
  cislo:=chr(43)+cislo;
  R:=strtofloat(cislo);
  if Y=7 then
  begin
    R:=R*0.0012;
    M[Y,Z]:=R;
  end;
  if Y=11 then
  begin
    R:=R*10;
    M[Y,Z]:=R;
  end;
  if Y=8 then
  begin
    R:=R*0.039;
    M[Y,Z]:=R;
  end;
  cislo:='';
  K:=3;
  R:=0;
  Y:=Y+1;
end;
if (I=10) then
begin
  Chart1.SeriesList.Series[1].AddXY(M[7,Z],M[11,Z]);
  Chart2.SeriesList.Series[1].AddXY(M[7,Z],M[8,Z]);
  Y:=6;
  Z:=Z+1;
end;
end;
CloseFile(Soubor);
end;
end;
if ComboBox2.Text = 'Schéma bez otáčkového omezovače' then
begin
begin
  Chart1.SeriesList.series[0].Clear;
  Chart2.SeriesList.series[0].Clear;
  Chart1.BottomAxis.Automatic:=false;
  Chart1.BottomAxis.Minimum:=0;
  Chart1.BottomAxis.Maximum:=1;
  Chart2.BottomAxis.Automatic:=false;
  Chart2.BottomAxis.Minimum:=0;
  Chart2.BottomAxis.Maximum:=1;
  Chart2.LeftAxis.AutomaticMaximum:=false;
  Chart2.LeftAxis.Maximum:=8;
  AssignFile(Soubor, 'Uf-s-' + ComboBox1.Text + '.txt');
  Reset(Soubor);
  cislo:='';
  K:=3;
  Y:=2;
  Z:=0;
  while not eof(Soubor) do
  begin
    Read(Soubor,C);

```

```

        if C='.' then
            C:=', ';
            I:=ord(C);
            if (((I>47) and (I<58))or (I=44)or (I=101) or (I=43) or (I=45)) and
((K=0)or (K=3)) then
                begin
                    cislo:=cislo+C;
                    K:=0;
                end;
            if ((I=32) and (K<>3)) then
                begin
                    Q:=ord(cislo[1]);
                    if Q<>45 then
                        cislo:=chr(43)+cislo;
                    R:=strtofloat(cislo);
                    M[Y,Z]:=R;
                    cislo:=' ';
                    K:=3;
                    R:=0;
                    Y:=Y+1;
                end;
            if (I=10) then
                begin
                    Chart1.SeriesList.Series[0].AddXY(M[2,Z],M[3,Z]);
                    Chart2.SeriesList.Series[0].AddXY(M[2,Z],M[4,Z]);
                    Y:=1;
                    Z:=Z+1;
                end;
            end;
        CloseFile(Soubor);
        Label7.Caption:=FloatToStr(M[5,3]);
        Label8.Caption:=FloatToStr(M[6,3]);
        Label9.Visible:=false;
        Label10.Visible:=false;
        Label11.Visible:=false;
        Label12.Visible:=false;
        Label13.Visible:=false;
        Label28.Caption:=FloatToStr(M[7,3])+ ' Nm';
    end;
begin
    Chart1.SeriesList.series[1].Clear;
    Chart2.SeriesList.series[1].Clear;
    AssignFile(Soubor, 'Uf-m-'+ComboBox1.Text+'.txt');
    Reset(Soubor);
    cislo:=' ';
    K:=3;
    Y:=1;
    Z:=1;
    while not eof(Soubor) do
        begin
            Read(Soubor,C);
            I:=ord(C);
            if (((I>47) and (I<58))or (I=46)or (I=43) or (I=45))and ((K=0)or (K=3)))
then
                begin
                    if I=46 then
                        C:=', ';
                        cislo:=cislo+C;
                        K:=0;
                    end;
                    if ((I=32) and (K<>3)) then
                        begin
                            Q:=ord(cislo[1]);
                            if Q<>45 then
                                cislo:=chr(43)+cislo;
                            R:=strtofloat(cislo);
                            if Y=7 then
                                begin
                                    R:=R*0.0012;
                                    M[Y,Z]:=R;
                                end;
                            if Y=11 then
                                begin
                                    R:=R*10;
                                    M[Y,Z]:=R;
                                end;
                            if Y=8 then

```

```

        begin
            R:=R*0.039;
            M[Y,Z]:=R;
        end;
        cislo:='';
        K:=3;
        R:=0;
        Y:=Y+1;
    end;
    if (I=10) then
        begin
            Chart1.SeriesList.Series[1].AddXY(M[7,Z],M[11,Z]);
            Chart2.SeriesList.Series[1].AddXY(M[7,Z],M[8,Z]);
            Y:=6;
            Z:=Z+1;
        end;
    end;
    CloseFile(Soubor);
end;
end;
end;

procedure TForm1.Button3Click(Sender: TObject);
type
    TSoubor = TextFile ;
    Matice = array [0..20,0..3000] of real;

var
    Soubor: TSoubor;
    C: char;
    R: real;
    M: Matice;
    Q,K,I,Y,Z: integer;
    cislo: string;

begin
    if not ((ComboBox3.Text = 'Schéma pro textilní aplikace')or
        (ComboBox3.Text = 'Schéma s otáčkovým a proudovým omezovačem')or
        (ComboBox3.Text = 'Schéma bez otáčkového omezovače'))then
        ShowMessage('Prosím zvolte simulační schéma');

    if ComboBox3.Text = 'Schéma pro textilní aplikace' then
        begin
            begin
                Chart3.SeriesList.series[0].Clear;
                Chart4.SeriesList.series[0].Clear;
                Chart3.BottomAxis.Automatic:=false;
                Chart3.BottomAxis.Minimum:=0;
                Chart3.BottomAxis.Maximum:=1;
                Chart4.BottomAxis.Automatic:=false;
                Chart4.BottomAxis.Minimum:=0;
                Chart4.BottomAxis.Maximum:=1;
                Chart4.LeftAxis.AutomaticMaximum:=false;
                Chart4.LeftAxis.Maximum:=8;
                AssignFile(Soubor, 'Uftex-s-'+ComboBox4.Text+'.txt');
                Reset(Soubor);
                cislo:='';
                K:=3;
                Y:=2;
                Z:=0;
                while not eof(Soubor) do
                    begin
                        Read(Soubor,C);
                        if C='.' then
                            C:=',';
                        I:=ord(C);
                        if (((I>47) and (I<58))or (I=44)or (I=101) or (I=43) or (I=45)) and
                            ((K=0)or (K=3))) then
                            begin
                                cislo:=cislo+C;
                                K:=0;
                            end;
                        if ((I=32) and (K<>3)) then
                            begin
                                Q:=ord(cislo[1]);
                                if Q<>45 then
                                    cislo:=chr(43)+cislo;
                                end;
                            end;
                    end;
                end;
            end;
        end;
    end;
end;

```



```

        R:=strtofloat(cislo);
        M[Y,Z]:=R;
        cislo:='';
        K:=3;
        R:=0;
        Y:=Y+1;
    end;
    if (I=10) then
    begin
        Chart3.SeriesList.Series[0].AddXY(M[2,Z],M[3,Z]);
        Chart4.SeriesList.Series[0].AddXY(M[2,Z],M[4,Z]);
        Y:=1;
        Z:=Z+1;
    end;
    end;
    CloseFile(Soubor);
    Label20.Caption:=FloatToStr(M[5,3]);
    Label21.Caption:=FloatToStr(M[6,3]);
    Label22.Visible:=false;
    Label23.Visible:=false;
    Label24.Visible:=false;
    Label25.Visible:=false;
    Label26.Visible:=false;
    Label30.Caption:=FloatToStr(M[7,3])+ ' Nm';
end;
begin
    Chart3.SeriesList.series[1].Clear;
    Chart4.SeriesList.series[1].Clear;
    AssignFile(Soubor, 'Uftex-m-'+ComboBox4.Text+'.txt');
    Reset(Soubor);
    cislo:='';
    K:=3;
    Y:=1;
    Z:=1;
    while not eof(Soubor) do
    begin
        Read(Soubor,C);
        I:=ord(C);
        if (((I>47) and (I<58))or (I=46)or (I=43) or (I=45))and ((K=0)or (K=3)))
then
            begin
                if I=46 then
                    C:',';
                    cislo:=cislo+C;
                    K:=0;
                end;
                if ((I=32) and (K<>3)) then
                begin
                    Q:=ord(cislo[1]);
                    if Q<>45 then
                        cislo:=chr(43)+cislo;
                    R:=strtofloat(cislo);
                    if Y=7 then
                        begin
                            R:=R*0.0012;
                            M[Y,Z]:=R;
                        end;
                    if Y=11 then
                        begin
                            R:=R*10;
                            M[Y,Z]:=R;
                        end;
                    if Y=8 then
                        begin
                            R:=R*0.039;
                            M[Y,Z]:=R;
                        end;
                    cislo:='';
                    K:=3;
                    R:=0;
                    Y:=Y+1;
                end;
                if (I=10) then
                begin
                    Chart3.SeriesList.Series[1].AddXY(M[7,Z],M[11,Z]);
                    Chart4.SeriesList.Series[1].AddXY(M[7,Z],M[8,Z]);
                    Y:=6;

```

```

        Z:=Z+1;
    end;
end;
CloseFile(Soubor);
end;
end;
if ComboBox3.Text = 'Schéma s otáčkovým a proudovým omezovačem' then
begin
    begin
        Chart3.SeriesList.series[0].Clear;
        Chart4.SeriesList.series[0].Clear;
        Chart3.BottomAxis.Automatic:=false;
        Chart3.BottomAxis.Minimum:=0;
        Chart3.BottomAxis.Maximum:=1;
        Chart4.BottomAxis.Automatic:=false;
        Chart4.BottomAxis.Minimum:=0;
        Chart4.BottomAxis.Maximum:=1;
        Chart4.LeftAxis.AutomaticMaximum:=false;
        Chart4.LeftAxis.Maximum:=8;
        AssignFile(Soubor, 'Ufirc-s-' + ComboBox4.Text + '.txt');
        Reset(Soubor);
        cislo:='';
        K:=3;
        Y:=2;
        Z:=0;
        while not eof(Soubor) do
            begin
                Read(Soubor,C);
                if C='.' then
                    C:=',';
                    I:=ord(C);
                    if (((I>47) and (I<58))or (I=44)or (I=101) or (I=43) or (I=45)) and
((K=0)or (K=3)) then
                        begin
                            cislo:=cislo+C;
                            K:=0;
                        end;
                    if ((I=32) and (K<>3)) then
                        begin
                            Q:=ord(cislo[1]);
                            if Q<>45 then
                                cislo:=chr(43)+cislo;
                                R:=strtofloat(cislo);
                                M[Y,Z]:=R;
                                cislo:='';
                                K:=3;
                                R:=0;
                                Y:=Y+1;
                            end;
                        if (I=10) then
                            begin
                                Chart3.SeriesList.Series[0].AddXY(M[2,Z],M[3,Z]);
                                Chart4.SeriesList.Series[0].AddXY(M[2,Z],M[4,Z]);
                                Y:=1;
                                Z:=Z+1;
                            end;
                        end;
                    CloseFile(Soubor);
                    Label20.Caption:=FloatToStr(M[5,3]);
                    Label21.Caption:=FloatToStr(M[6,3]);
                    Label22.Visible:=true;
                    Label23.Visible:=true;
                    Label24.Visible:=true;
                    Label25.Visible:=true;
                    Label26.Visible:=true;
                    Label25.Caption:=FloatToStr(M[7,3]);
                    Label26.Caption:=FloatToStr(M[8,3]);
                    Label30.Caption:=FloatToStr(M[9,3])+' Nm';
                end;
            begin
                Chart3.SeriesList.series[1].Clear;
                Chart4.SeriesList.series[1].Clear;
                AssignFile(Soubor, 'Ufirc-m-' + ComboBox4.Text + '.txt');
                Reset(Soubor);
                cislo:='';
                K:=3;
                Y:=1;

```

```

Z:=1;
while not eof(Soubor) do
begin
  Read(Soubor,C);
  I:=ord(C);
  if (((I>47) and (I<58))or (I=46)or (I=43) or (I=45))and ((K=0)or (K=3)))
then
begin
  if I=46 then
  C:=',';
  cislo:=cislo+C;
  K:=0;
end;
if ((I=32) and (K<>3)) then
begin
  Q:=ord(cislo[1]);
  if Q<>45 then
  cislo:=chr(43)+cislo;
  R:=strtofloat(cislo);
  if Y=7 then
  begin
    R:=R*0.0012;
    M[Y,Z]:=R;
  end;
  if Y=11 then
  begin
    R:=R*10;
    M[Y,Z]:=R;
  end;
  if Y=8 then
  begin
    R:=R*0.039;
    M[Y,Z]:=R;
  end;
  cislo:='';
  K:=3;
  R:=0;
  Y:=Y+1;
end;
if (I=10) then
begin
  Chart3.SeriesList.Series[1].AddXY(M[7,Z],M[11,Z]);
  Chart4.SeriesList.Series[1].AddXY(M[7,Z],M[8,Z]);
  Y:=6;
  Z:=Z+1;
end;
end;
CloseFile(Soubor);
end;
end;
if ComboBox3.Text = 'Schéma bez otáčkového omezovače' then
begin
begin
  Chart3.SeriesList.series[0].Clear;
  Chart4.SeriesList.series[0].Clear;
  Chart3.BottomAxis.Automatic:=false;
  Chart3.BottomAxis.Minimum:=0;
  Chart3.BottomAxis.Maximum:=1;
  Chart4.BottomAxis.Automatic:=false;
  Chart4.BottomAxis.Minimum:=0;
  Chart4.BottomAxis.Maximum:=1;
  Chart4.LeftAxis.AutomaticMaximum:=false;
  Chart4.LeftAxis.Maximum:=8;
  AssignFile(Soubor, 'Uf-s-'+ComboBox4.Text+'.txt');
  Reset(Soubor);
  cislo:='';
  K:=3;
  Y:=2;
  Z:=0;
  while not eof(Soubor) do
  begin
    Read(Soubor,C);
    if C='.' then
    C:=',';
    I:=ord(C);
    if (((I>47) and (I<58))or (I=44)or (I=101) or (I=43) or (I=45)) and
((K=0)or (K=3))) then

```

[illegible]

```

        K:=3;
        R:=0;
        Y:=Y+1;
    end;
    if (I=10) then
    begin
        Chart3.SeriesList.Series[1].AddXY(M[7,Z],M[11,Z]);
        Chart4.SeriesList.Series[1].AddXY(M[7,Z],M[8,Z]);
        Y:=6;
        Z:=Z+1;
    end;
    end;
    CloseFile(Soubor);
end;
end;
end;

procedure TForm1.BitBtn1Click(Sender: TObject);
begin
    case PageControl1.ActivePageIndex of
        0 : Chart1.print;
        1 : Chart2.print;
    end;
end;

procedure TForm1.BitBtn2Click(Sender: TObject);
begin
    case PageControl1.ActivePageIndex of
        0 : begin
            if ComboBox2.Text = 'Schéma pro textilní aplikace' then
                Chart1.SaveToBitmapFile('gr1_ot-tex-'+ComboBox1.Text+'.bmp');
            if ComboBox2.Text = 'Schéma s otáčkovým a proudovým omezovačem' then
                Chart1.SaveToBitmapFile('gr1_ot-irc-'+ComboBox1.Text+'.bmp');
            if ComboBox2.Text = 'Schéma bez otáčkového omezovače' then
                Chart1.SaveToBitmapFile('gr1_ot-'+ComboBox1.Text+'.bmp');
            end;
        1 : begin
            if ComboBox2.Text = 'Schéma pro textilní aplikace' then
                Chart1.SaveToBitmapFile('gr1_pr-tex-'+ComboBox1.Text+'.bmp');
            if ComboBox2.Text = 'Schéma s otáčkovým a proudovým omezovačem' then
                Chart1.SaveToBitmapFile('gr1_pr-irc-'+ComboBox1.Text+'.bmp');
            if ComboBox2.Text = 'Schéma bez otáčkového omezovače' then
                Chart1.SaveToBitmapFile('gr1_pr-'+ComboBox1.Text+'.bmp');
            end;
        end;
        ShowMessage('Program uložil otevřený graf do aktuálního adresáře')
    end;

procedure TForm1.FormCreate(Sender: TObject);
begin
    PageControl1.ActivePageIndex := 0;
    PageControl2.ActivePageIndex := 0;
end;

procedure TForm1.BitBtn3Click(Sender: TObject);
begin
    case PageControl2.ActivePageIndex of
        0 : begin
            if ComboBox3.Text = 'Schéma pro textilní aplikace' then
                Chart3.SaveToBitmapFile('gr2_ot-tex-'+ComboBox4.Text+'.bmp');
            if ComboBox3.Text = 'Schéma s otáčkovým a proudovým omezovačem' then
                Chart3.SaveToBitmapFile('gr2_ot-irc-'+ComboBox4.Text+'.bmp');
            if ComboBox3.Text = 'Schéma bez otáčkového omezovače' then
                Chart3.SaveToBitmapFile('gr2_ot-'+ComboBox4.Text+'.bmp');
            end;
        1 : begin
            if ComboBox3.Text = 'Schéma pro textilní aplikace' then
                Chart4.SaveToBitmapFile('gr2_pr-tex-'+ComboBox4.Text+'.bmp');
            if ComboBox3.Text = 'Schéma s otáčkovým a proudovým omezovačem' then
                Chart4.SaveToBitmapFile('gr2_pr-irc-'+ComboBox4.Text+'.bmp');
            if ComboBox3.Text = 'Schéma bez otáčkového omezovače' then
                Chart4.SaveToBitmapFile('gr2_pr-'+ComboBox4.Text+'.bmp');
            end;
        end;
        ShowMessage('Program uložil otevřený graf do aktuálního adresáře')
    end;
end;

```

```
procedure TForm1.BitBtn4Click(Sender: TObject);
begin
  case PageControl2.ActivePageIndex of
    0 : Chart1.print;
    1 : Chart2.print;
  end;
end;

end.
```